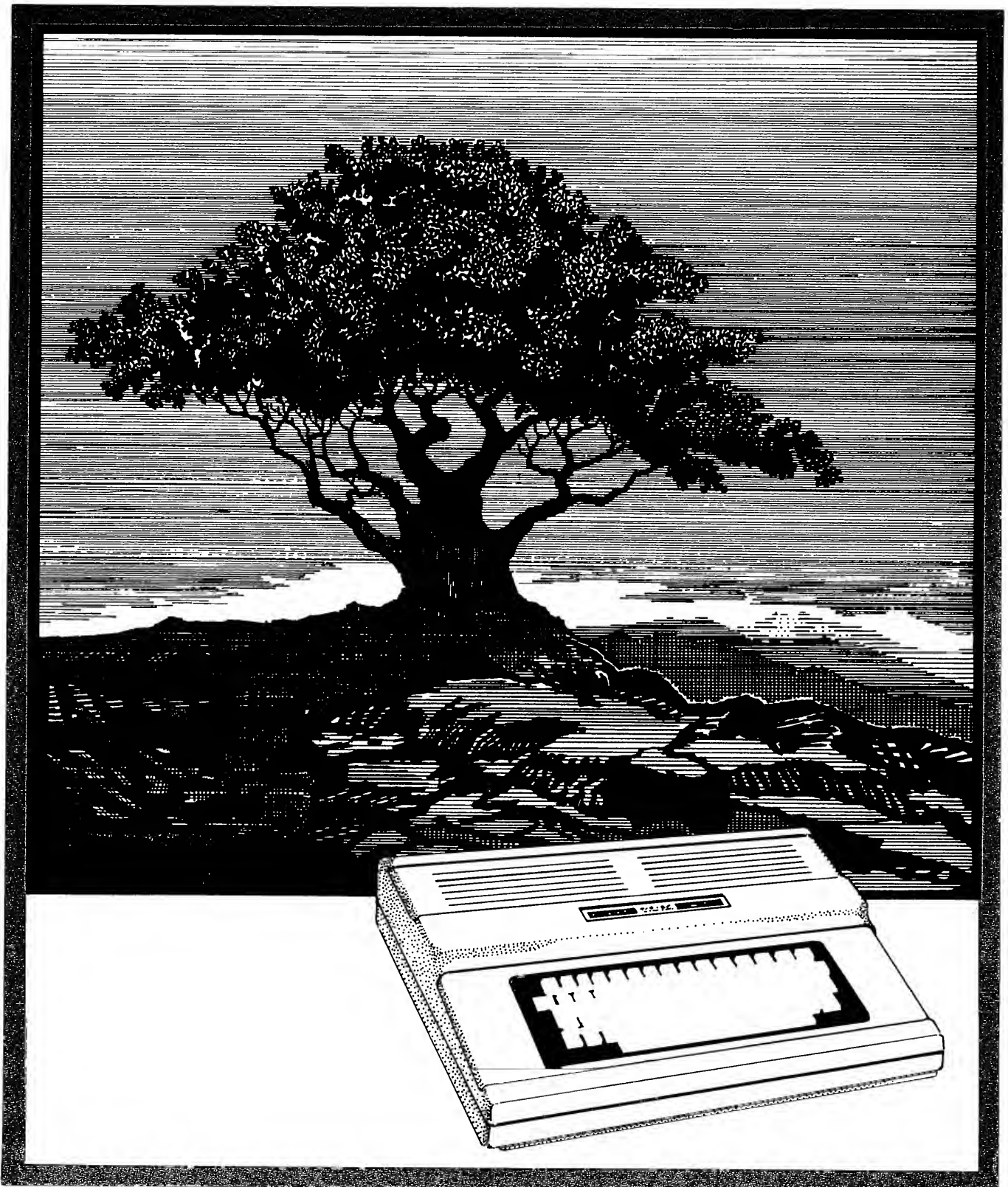


# UNDERCOLOR

Vol. 1 No. 9

April 22, 1985



FIRST CLASS MAIL  
POSTAGE PAID  
MANCHESTER, NH  
PERMIT #1926

\$2.00

# A SPECTRUM PROJECTS SHOPPING LIST

## A CHIP OFF THE OLD...

6821 Standard PIA .....\$9.95  
 68766 (Fits Disk Basic Skt) Eprom.\$24.95  
 27128 16K DOS Eprom (Or 2 DOS's!).\$24.95  
 New SAM Chip w/heatsink (74LS785).\$29.95  
 6809E CPU Chip .....\$29.95  
 Ext Basic 1.1 ROM - NEW LOW PRICE.\$29.95  
 28 pin Ext Basic upgrade 26-3134A.\$34.95  
 Basic ROM 1.2 Chip (30% FASTER) ..\$39.95  
 Disk ROM 1.1 (New DOS Command) ..\$39.95  
 Eprom Eraser - 3 min erasure time.\$49.95  
 Lower Kit Bd - Specify CoCoI/II\*\*.\$59.95  
 CoCo First Aid Kit - includes 2 PIAs,  
 6809E & SAM (Be Prepared!!!) ....\$59.95  
 26-3134A & 26-3136A 64K Upgrade - 2 chip  
 set (ONLY for new Korean CoCoIIA).\$69.95  
 Eprom Prgmr (2ms speed/2K - 16K).\$139.95  
 64K CoCo II - w/NEW keyboard ....\$169.95

## COCO LIBRARY...

CoCo Memory Map .....\$12.00  
 Rainbow Book & Tape of Adventures.\$14.95  
 Basic Programming Tricks Revealed.\$14.95  
 The FACTS - Inside "guts" of CoCo.\$14.95  
 500 Pokes, Peeks 'N Execs .....\$16.95  
 Basic 09 Tour Guide .....\$18.95  
 Disk Basic (1.0/1.1) Unraveled ...\$19.95  
 Color Basic 1.2 Unraveled .....\$19.95  
 New! CoCo II Service Manual .....\$19.95  
 The Complete Rainbow Guide to OS9.\$19.95  
 W/Two Disk Package of demo pgms ..\$49.95

## MORE GOOD STUFF...

27128/68764 Adapter - Allows you to  
 plug-in 28pin EPROMs into 24pin sockets!  
 Fits all controllers for 16K DOSs \$19.95  
 CoCo Freeze Frame - Stop your CoCo dead  
 in its tracks! Put games on "HOLD"\$19.95  
 CoCo Light Pen with 6 programs ...\$24.95  
 Spectrum Voice Pak - SAVE \$30!! ..\$39.95  
 PBH Parallel Interface - Beats Botek!  
 300-9600 baud w/ptr-modem switch .\$69.95  
 PBJ WORD-PAK II - Hi-Res 80x24 display  
 w/smooth scrolling & 8x10 matrix \$139.95  
 PBJ C-C Bus - 6 slot interface ..\$149.95  
 NOMAD - The CoCo Robot .....\$199.95

All orders plus \$3.00 S/H (Foreign \$5.00)  
 COD add \$2.00 extra  
 NYS Residents add Sales Tax

## COCO CABLES AND...

Printer/Modem 15' Extender Cable ..\$14.95  
 Tired of unplugging devices from your  
 RS232 port? Try a RS232 "Y" Cable.\$19.95  
 Joystick/Mouse 10' Extender Cable.\$19.95  
 Null Modem Cable - 40 pin to DB25.\$24.95  
 Disk Interface/Rom Pak Extender - Move  
 your disks & ROM Paks (3 feet) ..\$29.95  
 Triple RS232 Switcher - Now select one  
 of any three RS232 peripherals ...\$29.95  
 40 Pin Dual "Y" Cable - Hook up a Disk  
 w/Voice, Word Pak, CoCo Max,etc ..\$29.95  
 Triple "Joyport" Switcher - Joystick,  
 Mouse, Touch Pad and/or Light Pen \$39.95  
 40 Pin Triple "Y" Cable - Hook up any 3-  
 Voice/Word/RS232/Digitizer PAKs ..\$39.95  
 Finally! 24" Multi-Pak Extender ..\$39.95

## OTHER GOOD STUFF...

C-10 tapes in any quantity .....49 cents  
 5 1/4" Diskettes in any quantity ..\$1.49  
 Joystick, Cassette or Serial plug ..\$3.99  
 32K, 64K or 128K RAM Button .....\$4.99  
 Rompak w/Blank PC Brd 27xx series ..\$9.95  
 The Disk Doubler - Doubleside your 5 1/4  
 diskettes for 160K more storage ..\$19.95  
 Video Clear - This cable will reduce TV  
 interference created by CoCo! ....\$19.95  
 Video Reverser -Reduce eyestrain w/green  
 letters on black background\*\* ....\$24.95  
 The Magic Box - Load Mod I/III Basic  
 program tapes into the CoCo .....\$24.95  
 DOS Switcher - Select from any two DOSs  
 (Disk 1.0 1.1, JDOS) in J&M ctrlr .\$24.95  
 Stereo Pak - Hardware synthesizer used  
 w/Musica 2...superb stereo sound! \$39.95  
 CoCo Cooler- State D,E or CoCoII ..\$49.95  
 MicroWorks Video Digitizer .....\$149.95  
 \*\* N O T for 26-3134/36 CoCo II's

**SPECTRUM PROJECTS**  
**PO BOX 21272**  
**93-15 86TH DRIVE**  
**WOODHAVEN NY 11421**

**718-441-2807**

**SPECIAL 64K RAM CHIPS \$29.95**

## Who's Who

### Editor

Debra A. Marshall

### Technical

Dennis Bathory Kitsz

### Contributing Photographer

Charley Freiberg

### Production Manager

Clare McCarthy

### Paste-Up

Janet Patterson

### Logistics Coordinator

JoAnn Trotter

### Technical Consultant and Publisher

Dennis Kitsz

### Advertising:

Judy Knapp  
(802)485-6139

### Advertising:

Spencer Knowlton  
(207)785-4151

### Office

(802)485-6440

### Editorial Office

(207)785-5148

Published By: ColorPlus Inc., Box 6809, Roxbury, VT 05669. (802)485-6440.

**Circulation Problems?** 2nd Class permit pending. Contact David Gunn, Box 6809, Roxbury, VT 05669, 802-485-6440.

**Subscriptions:** United States 1 year—\$33; 2 years—\$66. First class mail 1 year—\$47; 2 years—\$94. Canada and Mexico 1 year \$43; 2 years \$86. Overseas 1 year \$53 (US dollars); Airmail \$65 (US dollars). UnderColor is published 26 times a year. Call (802)485-6440, or write to UnderColor, Box 6809, Roxbury, VT 05669, Attn: Perley Swann.

**Advertisers:** rates available.

**Editorial Office:** c/o Debra Marshall, RFD 1, Box 786, Union, ME 04862, 207-785-5148. Submissions are welcome; include S.A.S.E. for returns. To receive author's guide, write: Author's Guide, Box 6809, Roxbury, VT 05669. Letters to Editor, manuscripts, article queries, etc. should be sent to editorial office.

# Table of Contents

- **Defusr** ..... page 2  
By Terry Kepner *Q&A.*
- **Graphics** ..... page 4  
By Dale Keller *Neat graphics.*
- **Fie On Intel Hex** .... page 12  
By Stephen P. Allen *Fixing hex's numbers.*
- **Off Color: Gampak 3-2-1**  
..... page 16  
By Larry Cadman *The ultimate package.*
- **Compress That Format**  
..... page 18  
By Richard Ramella *Squash it, flatten it, fit it.*
- **Errors and Omissions**  
..... page 21  
*Ok, we found it – Figure 4 of Real-Time Clock!*
- **Reviews** ..... page 23

*I made a password program for my TRS-80 Color Computer 2:*

```

5 CLS
6 DATA "INSERT ANY WORD YOU WANT"
7 FOR X=1 TO 1
8 READ F$
9 INPUT "ENTER CODE";B$
10 IF B$=F$ THEN 20
11 IF B$<>F$ THEN 401
12 END
20 main program
    /
    /
    /
400 END
401 CLS
405 PRINT@100, "WRONG CODE"
406 FORX=1 TO 200:NEXT
410 CLS(4)
415 SOUND 150,4
420 GOTO 410

```

*The program works OK but anyone who wants to get in can list the password program and look for the code word. What do you suggest I do to stop this without physically having to stop the person? Is there something I can add to the program?*

*No name or address*

Put this command in your program: POKE 383,158. It disables the list command. People can stop the program, but they can't read it. POKE 383,0 restores the list command.

*I have a 16K CoCo, rev. N.C., and want to install (piggyback) another 16K. Where do I tie the pin 4 bus on the upper bank of chips to make it work? I tried going to pin 35 of the 6883 as I did on a Rev. E board but it won't work. Do I need to cut a trace? Move jumpers? What?*

Dave Reichert  
Corona, CA

You made a mistake—it should work just fine. But you'd be better off putting in 64K RAMs. The modifications you want to do won't work with programs that move the video display page into the higher memory. The modification messes up the timing signals from the SAM.

*I own a 64K Color Computer. Recently I acquired a disk drive 0 and Disk Color Scribes. I'm quite pleased with the program except for one small problem. I type much faster than the Scribes program does. Though the memory remembers some of what I type in, it usually ends up getting jumbled. My typing speed is around 75 wpm. Is there any way to make Scribes's "typing" keep up with mine? Even if I could speed it up a little it would help.*

Janice M. Hill  
Harrison, NJ

I can't help you, I don't know of any way to speed up Color Scribes. Sorry.

*My wife gave me a Radio Shack Color Computer Editor Assembler as a gift. Since then I've gone through all kinds of books and I'm still stumped on the same old problem. Every book describes the commands in detail but not one takes the reader through a short, simple program and shows how it is used.*

*Personally, I like to see the results of a program on my monitor. The sample program in the Editor is a perfect example of what I'm looking for. I've played with that program, changed it, modified it until I can't think of anything else I can do to it. Can you help?*

W.J. Andrews  
Palm Bay, FL

I suggest you get Bill Barden's assembly language programming book for the Color Computer. It should be available at your Radio Shack store. It's the easiest one I've found.

You should also watch the Color Computer magazines, they frequently have A/L articles for the beginning programmer.

*I recently sold my 64K CoCo2 and happily purchased a Dragon 64 instead. The difference between the two is slim and Dragon's extras, including the price (\$149.95), convinced me that the Dragon is the superior computer.*

*The only question in my mind is the language compatibility and also the peripherals (modem, disk controller, etc.) that work with both computers. Just last week I happily purchased a Modem IB from Radio Shack and was upset not to know for certain that it would work on my Dragon 64. Could you help me out here?*

Basic is now not a problem with the Dragon Cruncher (Elkan Electronics) but some assembly language programs might cause a slight problem. Is there no way to accomplish this seemingly impossible task?

*Could you give me your personal recommendation on two things? First, what is your clear choice between the Coco and the Dragon 64? Second, what is the best terminal package for the money?*

Dave Chaplin  
Erving, MA

The modem should work just fine with the Dragon; all you need is the cable to connect the two.

The A/L compatibility problem can't be fixed by the average person. It requires that you re-write the keyboard scan routine so it uses the Dragon keyboard instead of that of the Color Computer.

Now that Tano is gone, the only choice is the Color Computer. While the Dragon has superior features, the compatibility problem is too annoying.

I like Colorcom/E, but VIP terminal is a real close second.

*I have a 16K Extended Basic Color Computer. When I try to use the joysticks in one of the Pmodes,*

the figure stays in the upper left of the screen. How can I get it on the rest of the screen?

Here is the program I use:

```
5 PMODE 3,1:SCREEN 1,1:PCLS
10 A=JOYSTK(0):B=JOYSTK(1)
20 PSET (A,B,4)
30 GOTO 10
```

Paul Calclease

If you have any games that use the joysticks and they work, I would say you are using the wrong joystick port. Try the joystick in the other port. If that doesn't work, change the program to use the other port (2 and 3) and try the joystick in both ports. If neither work and games that use those ports also don't work, then you probably have a blown PIA chip. Take your computer in for repairs.

I have recently upgraded my Color Computer to 64K. How do you go about switching between different banks of memory? Also, can you switch either the lower 32K addresses or the high 32K, or is it restricted to one portion of memory? My system contains one disk drive.

Thomas Rutledge  
Brookhaven, MS

You can't switch between banks in the manner you propose. The RAM is set up by the SAM chip as either a 32K RAM-32K ROM machine or an all-RAM 64K machine. If you switch out the ROM, you

lose all control because the 6809 chip no longer has instructions available to tell it how to control the keyboard and video. The computer just locks up.

64K assembly language programs deal with this by including their own code that tells the 6809 how to run the computer.

I purchased a TRS-80 Color 2, Extended Basic 64K and when I am not trying to program the computer, I like to amuse myself with a game or two. I like the new 3D graphics. I can never find enough computer games to stimulate my mind.

When I ask Radio Shack about any compatible games, they say maybe. Do you know if I can buy a computer game for another machine, especially cassette form, that will play on my computer?

Joseph R. Szathmary  
Jacksonville, FL

As yet, you can't use the tapes from any other computer in your Color Computer. However, if it's games you want, write to Mark Data Products (24001 Alicia Pkwy., No. 207, Mission Viejo, CA 92691, 714-768-1551) and Tom Mix Software (4285 Bradford N.E., Grand Rapids, MI 49506, 616-957-0444) for their catalogs. Between the two of them they have the most and the best games, both adventure and arcade; and many use 3-D type graphic displays.



## **SUPERB GRAPHICS UTILITIES FOR YOUR COCO !!!**

**GRAFX NO Y-CABLES or MULTIPACKS NEEDED EVER!!**

### **VIZDRAW...**

SIMILAR TO OTHER COMPUTER'S PRINT SHOP AND MACPAINT! YOU CAN DESIGN PRACTICALLY A WHOLE SHEET OF PAPER IN THE COLOR COMPUTER'S MEMORY (7'X7"). MAKE PRESENTATION QUALITY GRAPHICS, SPECIAL NEWSLETTERS WITH OVER 30 DIFFERENT FONT SETS!! REQUIRES 64K JOYSTICKS OR MOUSE. RUNS ON DISK OR TAPE SYSTEMS. PRINTER OPTIONS FOR VIZDRAW ARE RS LP 8 FORMAT, EPSON, C-ITON, BENINI 10-10X.

**SPECIAL....ORDER NOW AND GET OPTIONAL FONTS SETS FREE!!**

**PLEASE SPECIFY DISK or TAPE....only \$49.95 +\$2 postage**

### **VIDX...VIDXD DIGITIZER**

DIGITIZE ANY VIDEO SOURCE VIA THE RS-232 PORT. SYSTEM REQUIREMENTS 16K TAPE, 32K TAPE OR DISK. SAVE HOURS DRAWING...DIGITIZE IT! EASY ONE STEP MENU! READS/WRITES TO GRAPHICON.

**VIDX-1 only \$99.95 + \$3 postage**

**VIDX-2 only \$124.95 + \$3 postage (VIDX-2 has 2 port RS-232 switcher)**

GRAFPAX.....joystick/mouse interface  
Access all pixels (256x192) via the RS-232 port. Draw those circles and arcs free-hand, even sign your name!! Digitize a picture (VIDX digitizer is optional) and touch it up without going into any other graphics utility! GRAFPAX is easily modifiable to any program with joystick/mouse readouts...such as those that us A-D converters. Comes with its own great graphics program.  
REQUIRES 64K DISK.....only \$59.95 + \$2 postage

**NEW!**

### **DSS PERIPHERALS DISK CONTROLLER**

Gold contacts and write/data L.E.D.

With RS disk ROM 1.1.....reg. \$139....special \$110

without disk rom.....only \$100 +\$2 postage

### **CERTRON DISKETTES**

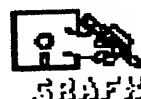
With plastic disk case and jackets, protect labels

SS/DD....\$12.50 DS/DD.....\$20.00

plus \$2 postage

64K RAM CHIPS....set of 8 w instructions only \$19.95 +\$2 postage  
We are now carrying most parts for the RS Disk CONTROLLERS  
Drop us a card for details and prices!!!!

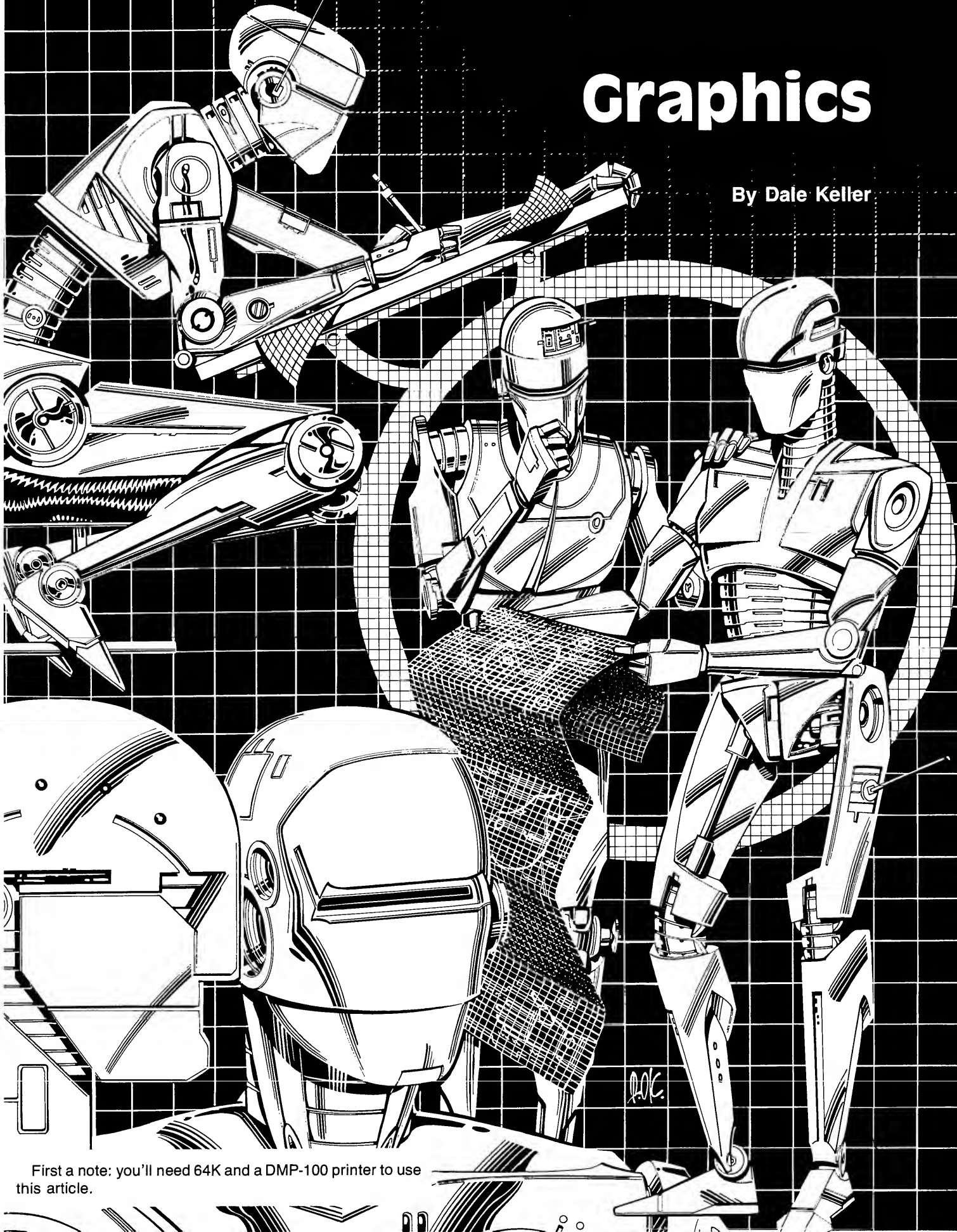
TO PLACE YOUR ORDER SEND CHECK OR MONEY ORDER TO:



P.O. BOX 254  
WEST MIFFLIN, PA 15122  
(412) 466-6974 (6-9 pm)  
COCONET 24 HR BBS 300/1200  
BAUD MODEM ONLINE  
(412) 744-2335  
DOWNLOAD FREE DIGITIZED PIX

# Graphics

By Dale Keller



First a note: you'll need 64K and a DMP-100 printer to use this article.



There is a use for that extra bank of 32K; you can increase your resolution to 768 by 384! This program lets you look at one-sixth of this big picture at a time, but you can use all of Basic's graphics commands on each screen. Additional commands are HLINE, HSET, and HPOINT (equivalent to LINE, PSET, and PPOINT, but using the entire picture coordinates). An INVERT command inverts each pixel on all six screens; to view the whole picture, PRINT prints all six screens together on a DMP-100 or equivalent printer (see Fig. 1). You can use PMODE 3 or 4; PMODE 3 gives you 384 x 384 resolution in four colors, though the colors show up only as shades of gray on the printer.

I originally wrote my work programs in assembly language using a CCEAD assembler from Eigen Systems, then later transferred it to EDTASM+. The data statements in the Basic program were taken from the CCEAD assembly, so some of the opcodes are slightly different. Listing 1 is a Basic program containing the assembly language program in data statements. The Basic program is actually two programs, an assembly language loader and the subroutine calls start at line 10000. A graphics editor, written to demonstrate the use of hi-res subroutines, runs from line 50-1170.

The graphics editor is an easy to use, menu driven program. Simply load Listing 1 and run it. When the main menu appears, select the option you want by hitting the appropriate key. Options include:

- Alter Default Values—you may select the color set to use, the PMODE, the color with which to draw, and the texture with which to paint (more on this later).
- Invert—inverts each pixel color on all six screens.
- Print—prints all six screens together as one picture on a printer. (This takes about half an hour.)
- File—allows you to save or load all six screens on a cassette recorder. (This takes about five minutes.)
- 1-6—display the selected screen.
- Draw Menu—displays the list of drawable objects.
- Hitting M from any menu or screen will return you to the main menu.

To draw an object, select the Draw menu option. Hitting D from any screen or the main menu will display the Draw menu.

The Basic graphics commands accept coordinates in the range of 0-255 (horizontal) and 0-191 (vertical) and draw on the current screen. The new (HLINE, HSET, HPOINT) accept coordinates in the range of 0-767 (horizontal) and 0-383 (vertical) and draw on the entire picture. The screens will flash by as HLINE and HSET execute, stopping on the screen on which the last pixel is to be set. Since the hi-res commands were intended for use on PMODE 4, the working color must be 0 (black) or 1 (white).

If you cause an error and the program breaks, you can restart it without losing your drawing by typing GOTO 50.

One more command, TEXTURE, has been added. This command acts very much like Paint, except that Texture colors the screen with a pattern instead of a solid color. This pattern is a number (selected from the Alter Defaults menu) which has no particular logical relation to the pattern drawn. You should experiment with this command. I used the technique because of its simplicity. Simply POKE 178,number and then PAINT (W,Y),1. See reference at end of article.

### Adapting

To use the graphics commands with your own program delete lines 40-1170. Leave line 30, which allocates memory and loads the assembly language code. Your program can then be written starting at line 50-9999.

The set-up routine at line 10000 clears the second bank of 32K which contains screens 2-6, but your program must clear screen number 1. Each subroutine has a comment stating what variables must be supplied for it to execute.

The print routine is set up for a DMP-100 printer at 1200 baud. Protocol for this printer is: CHR\$(18) sets the graphics mode. CHR\$(30) returns it to character mode. In each byte sent, the least significant bit is the top dot and the most significant bit is set to 1. If your printer does not use these commands, you will have to write your own print routine. Fig. 1 is a sample of what this program can do. The Basic program given will run on a cassette system. To use a disk system, type in the changes in Listing 2.

### References:

The texture technique is from Jim Hall's article in October '83 *Rainbow* magazine titled "The Pattern Poker." (end)

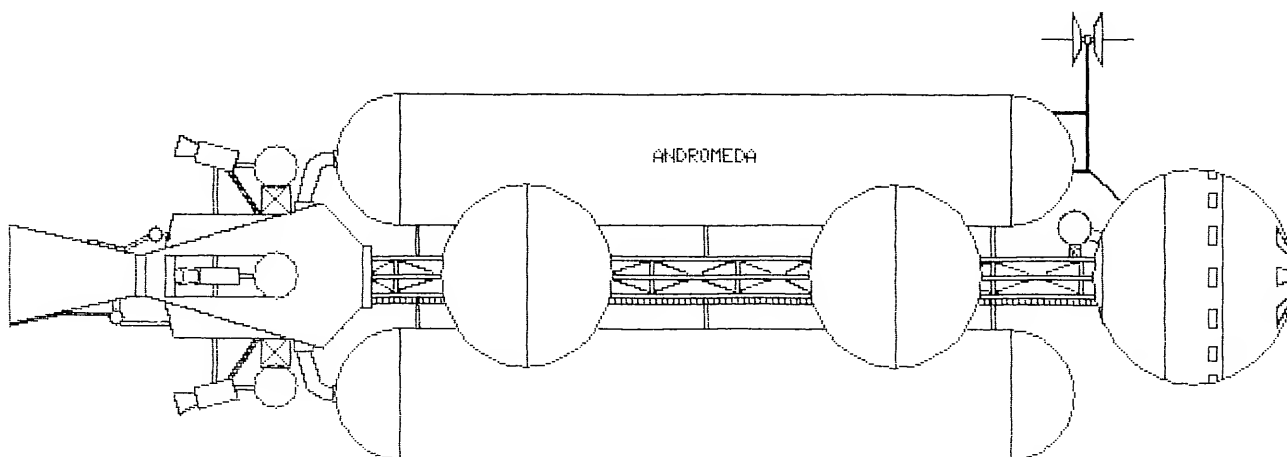


Figure 1. Sample

## Listing 1. Six Screen Graphics

```

10 'SIX SCREEN BASIC PROGRAM
15 'CASSETTE VERSION
20 'DALE A. KELLER 1984
30 GOTO 19000
40 '
50 'SET DEFAULTS
60 PM=4:WC=1:CS=1:TX=1
70 '
80 'MAIN MENU
90 CLS
100 PRINT"          MAIN MENU"
110 PRINT"A      = ALTER DEFAULT VALUES"
120 PRINT"D      = DRAW MENU"
130 PRINT"I      = INVERT ALL SCREENS"
140 PRINT"P      = PRINT ON PRINTER"
150 PRINT"F      = FILE (LOAD OR SAVE)"
160 PRINT"M      = MAIN MENU"
170 PRINT"1-6    = DISPLAY SCREEN 1-6"
180 '
190 A$=INKEY$:IF A$="" THEN 190
200 IF A$="A" THEN 310
210 IF A$="D" THEN 480
220 IF A$="I" THEN GOSUB 16000
230 IF A$="P" THEN GOSUB 17000:GOTO 80
240 IF A$="F" THEN 1090
250 IF A$="M" THEN 80
260 A=ASC(A$)-48
270 IF A>0 AND A<7 THEN SWAP=A:GOSUB 13000:PMODE P
M,1:SCREEN 1,CS
280 GOTO 190
290 '
300 '
310 'ALTER DEFAULTS
320 CLS:PRINT"          SET DEFAULTS MENU"
330 PRINT:PRINT"1 = WORKING COLOR (0-8)      ",WC
340 PRINT"2 = PMODE (3 OR 4)      ",PM
350 PRINT"3 = COLOR SET (1 OR 0)      ",CS
360 PRINT"4 = TEXTURE (0 - 255)      ",TX
370 PRINT:PRINT:M = MAIN MENU"
380 A$=INKEY$:IF A$="" THEN 380
390 '
400 IF A$="1" THEN INPUT"WORKING COLOR",WC:IF WC<0
OR WC>8 THEN 400
410 IF A$="2" THEN INPUT"PMODE",PM:IF PM<3 OR PM>4
THEN 410
420 IF A$="3" THEN INPUT "COLOR SET",CS:IF CS<0 OR
CS>1 THEN 420
430 IF A$="4" THEN INPUT "TEXTURE",TX:IF TX<0 OR T
X>255 THEN 430
440 IF A$="M" THEN 80
450 GOTO 310
460 '
470 '
480 'DRAW MENU
490 CLS:PRINT"          DRAW MENU"
500 PRINT:PRINT"1 = LINE"
510 PRINT"2 = HLINE"
520 PRINT"3 = PAINT"
530 PRINT"4 = CIRCLE"
540 PRINT"5 = PSET"
550 PRINT"6 = HSET"
560 PRINT"7 = PPOINT"
570 PRINT"8 = HPOINT"
580 PRINT"9 = PCLS"
590 PRINT"B = BOX"
600 PRINT"T = TEXTURE"
610 PRINT"M = MAIN MENU"
620 PRINT"D = DRAW MENU"
630 '
640 A$=INKEY$:IF A$="" THEN 640
650 IF A$="B" THEN CLS:PRINT"BOX":INPUT "X1,Y1,X2,
Y2",X1,Y1,X2,Y2:PMODE PM,1:SCREEN 1,CS:LINE(X1,Y1)-
(X2,Y2),PSET,B:GOTO 190
660 IF A$="T" THEN CLS:PRINT"TEXTURE":INPUT "X,Y,B
ORDER",X1,Y1,BR:PMODE PM,1:SCREEN 1,CS:POKE 178,TX:
PAINT (X1,Y1),BR:GOTO 190
670 IF A$="M" THEN 80
680 A=ASC(A$)-48
690 IF A<1 OR A>9 THEN 640
700 ON A GOTO 730,770,810,850,890,930,970,1010,106
0
710 '

```



```

720 '
730 'LINE
740 CLS:PRINT"LINE":INPUT "X1,Y1,X2,Y2";X1,Y1,X2,Y
2:PMODE PM,1:SCREEN 1,CS:COLOR WC:LINE(X1,Y1)-(X2,Y
2),PSET:GOTO 190
750 '
760 '
770 'HLINE
780 CLS:PRINT"HLINE":INPUT "X1,Y1,X2,Y2";X1,Y1,X2,
Y2:PMODE PM,1:SCREEN 1,CS:CO=WC:GOSUB 15000:GOTO 19
0
790 '
800 '
810 'PAINT
820 CLS:PRINT"PAINT":INPUT "X,Y,C1,C2";X1,Y1,C1,C2
:PMODE PM,1:SCREEN 1,CS:PAINT(X1,Y1),C1,C2:GOTO 190
830 '
840 '
850 'CIRCLE
860 CLS:PRINT"CIRCLE":INPUT "X,Y,R,C,HW,ST,EN";X1,
Y1,R,C,HW,ST,EN:PMODE PM,1:SCREEN 1,CS:CIRCLE(X1,Y1
),R,C,HW,ST,EN:GOTO 190
870 '
880 '
890 'PSET
900 CLS:PRINT"PSET":INPUT "X,Y";X1,Y1:PMODE PM,1:S
CREEN 1,CS:COLOR WC:PSET(X1,Y1,WC):GOTO 190
910 '
920 '
930 'HSET
940 CLS:PRINT"HSET":INPUT "X,Y";X1,Y1:PMODE PM,1:S
CREEN 1,CS:CO=WC:GOSUB 14000:GOTO 190
950 '
960 '
970 'PPOINT
980 CLS:PRINT"PPOINT":INPUT "X,Y";X1,Y1:PRINT"POIN
T IS COLOR"PPOINT(X1,Y1):GOTO 190
990 '
1000 '
1010 'HPOINT
1020 CLS:PRINT"HPOINT":INPUT "X,Y";X1,Y1:GOSUB 180
00:PRINT"POINT IS COLOR"PNT:GOTO 190
1030 '
1040 '
1050 'PCLS
1060 CLS:INPUT "ARE YOU SURE YOU WANT TO CLEAR TH
IS SCREEN";A$: IF LEFT$(A$,1)="Y" THEN PCLS:PMODE P
M,1:SCREEN 1,CS:GOTO 190:ELSE PMODE PM,1:SCREEN 1,C
S:GOTO 190
1070 '
1080 '
1090 'FILE MENU
1100 CLS:PRINT" FILE MENU"
1110 INPUT "SAVE OR LOAD";A$
1120 IF LEFT$(A$,1)="S" THEN GOSUB 11000
1130 IF LEFT$(A$,1)="L" THEN GOSUB 12000
1140 GOTO 80
1150 END
1160 '
1170 '
10000 'SETUP SYSTEM
10010 CLS:PRINT"PRINT" LOADING ML CODE"
10020 FOR T=&H7C55 TO &H7F47
10030 READ D
10040 POKE T,D
10050 NEXT T
10060 DATA183,255,214,26,80,183,255,223,142,128,0,
111,128,140,255,0,38,249,134,1,183,124,68,183,255,2
22,28,175,57,182,124,68,129,1,38,12,182,124,65,129,
1,39,29,183,124,68,32,30,246,124
10070 DATA65,247,124,67,183,124,65,189,124,163,134
,1,183,124,68,182,124,67,129,1,38,1,57,183,124,65,3
2,214,183,255,214,26,80,183,255,223,246,124,65,192,
2,79,31,1,79,95,140,0,0,39
10080 DATA7,195,24,0,48,31,38,249,195,128,0,31,2,1
42,6,0,166,0,230,32,167,160,231,128,140,30,0,38,243
,183,255,222,28,175,57,134,254,183,0,111,183,255,21
4,134,13,189,162,130,134,18
10090 DATA189,162,130,142,0,0,191,124,57,191,124,6
1,142,1,127,191,124,59,191,124,63,142,0,0,191,124,7
2,142,0,6,191,124,76,254,124,72,189,125,183,246,124
,71,182,124,74,86,70,51,65,17
10100 DATA131,0,7,39,22,183,124,74,190,124,61,48,1

```

```

191,124,61,191,124,57,190,124,63,191,124,59,32,215
,28,254,70,190,124,61,140,2,255,45,2,71,71,138,128,
189,162,130,16,190,124,63,49
10110 DATA63,16,191,124,63,16,191,124,59,16,140,25
5,255,39,14,252,124,61,179,124,76,253,124,61,253,12
4,57,32,160,190,124,61,48,1,191,124,61,191,124,57,1
6,142,1,127,16,191,124,63,16,191
10120 DATA124,59,134,13,189,162,130,190,124,61,140
,2,250,46,3,22,255,121,140,3,0,39,15,206,0,2,255,12
4,72,142,0,4,191,124,76,22,255,101,134,13,189,162,1
30,134,30,189,162,130,134,0
10130 DATA183,0,111,57,189,125,248,252,124,57,71,8
6,71,86,71,86,253,124,69,252,124,59,88,73,88,73,88,
73,88,73,88,73,243,124,69,195,6,0,31,1,252,124,57,1
96,248,253,124,69,252,124
10140 DATA57,179,124,69,203,1,166,0,73,90,38,252,7
3,132,1,183,124,71,57,134,1,190,124,57,140,1,0,45,5
,48,137,255,0,76,140,1,0,45,5,48,137,255,0,76,16,19
0,124,59,16,140
10150 DATA0,192,45,7,76,76,76,49,169,255,64,16,191
,124,59,191,124,57,177,124,68,39,6,183,124,65,189,1
24,114,57,189,125,248,252,124,57,71,86,71,86,71,86,
253,124,69,252,124,59,88,73
10160 DATA88,73,88,73,88,73,88,73,243,124,69,195,6
,0,31,1,252,124,57,196,248,253,124,69,252,124,57,17
9,124,69,131,0,7,80,134,1,193,0,39,4,90,72,32,248,2
46,124,82,39,10,183
10170 DATA124,65,166,0,186,124,65,32,9,67,183,124,
65,166,0,180,124,65,167,0,57,127,124,83,127,124,84,
252,124,61,179,124,57,253,124,72,42,5,134,255,183,1
24,83,252,124,63,179,124,59,253
10180 DATA124,74,42,5,134,255,183,124,84,252,124,5
7,88,73,88,73,253,124,76,127,124,78,252,124,59,88,7
3,88,73,253,124,79,127,124,81,189,126,53,252,124,72
,243,124,77,253,124,77,182,124,83
10190 DATA185,124,76,183,124,76,252,124,74,243,124
,80,253,124,80,182,124,84,185,124,79,183,124,79,252
,124,76,71,86,71,86,253,124,57,252,124,79,71,86,71,
86,253,124,59,16,179,124,63,38,191
10200 DATA252,124,57,16,179,124,61,38,182,189,126,
53,57,142,6,0,236,0,67,83,237,129,140,30,0,38,245,1
83,255,214,26,80,183,255,223,142,128,0,236,0,67,83,
237,129,140,248,0,38,245,183
10210 DATA255,222,28,175,57
10220 '
10230 'CLEAR
10240 DEFUSR0=&H7C55
10250 'GET
10260 DEFUSR1=&H7C72
10270 'SET/RESET
10280 DEFUSR2=&H7E35
10290 'LINE
10300 DEFUSR3=&H7E90
10310 'PRINTOUT
10320 DEFUSR4=&H7CDC
10330 'INVERT
10340 DEFUSR5=&H7F1E
10350 'POINT
10360 DEFUSR6=&H7DB7
10370 X=USR0(X) 'CLEAR 5 SCREENS
10380 PMODE 4,1:PCLS 'CLEAR 1ST SCREEN
10390 GOTO 50
10400 '
10410 '
11000 'SAVE 6 SCREENS TO CASSETTE
11010 INPUT "FILE NAME";F$:INPUT "CASSETTE ON";A$

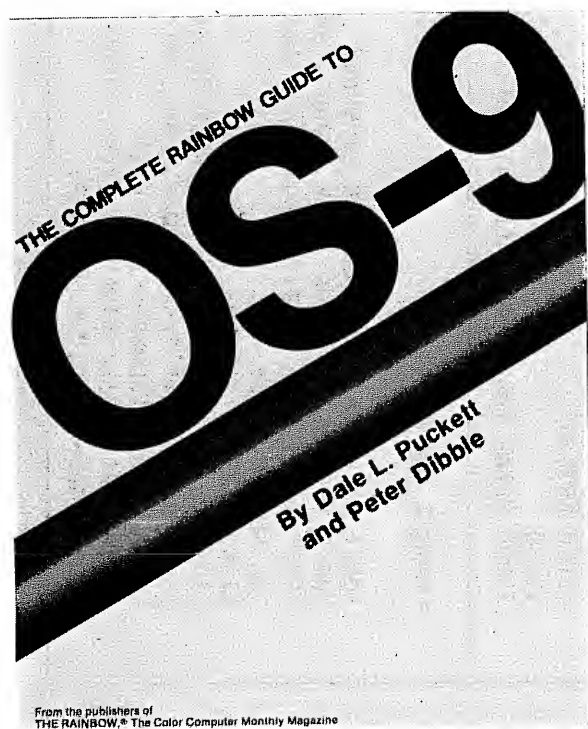
11020 FOR SWAP=1 TO 6
11030 POKE &H7C41,SWAP
11040 X=USR1(X)
11050 CSAVEM F$,1536,7679,&H7C71
11060 NEXT SWAP
11070 RETURN
11080 '
11090 '
12000 'LOAD 6 SCREENS FROM CASSETTE
12010 INPUT "FILE NAME";F$:INPUT "CASSETTE ON";A$
12020 FOR SWAP=1 TO 6
12030 POKE &H7C41,SWAP
12040 X=USR1(X)
12050 CLOADM F$
12060 NEXT SWAP
12070 POKE &H7C41,1
12080 X=USR1(X)
12090 RETURN
12100 '

```

More than a book . . .

## A MILESTONE

# THE COMPLETE RAINBOW GUIDE TO OS-9



Today's programmers use short modules of readable code to build complex programs. The OS-9 operating system and the high level languages it brings you make the job easy.

OS-9 has so many things going for it that you need a guide as comprehensive and thorough as *The Complete Rainbow Guide To OS-9* to show you how to talk to OS-9 and realize the potential of this extremely efficient implementation of the UNIX operating philosophy.

Co-authored by Dale L. Puckett and Peter Dibble — two of the foremost authorities on OS-9 — *The Complete Rainbow Guide to*

OS-9 demystifies the dynamic operating system that gives the Color Computer more power and flexibility than many of the high- cost computers on the market . . . and gives you the ability and confidence to reach new programming heights.

With *The Complete Rainbow Guide To OS-9*, you will be prepared to take full advantage of the multitasking system that is setting new standards for Color Computer programming. For only \$19.95!

## Also Available!

*The Rainbow Guide To OS-9 Disk*. An adjunct to and complement to the book. You'll want the book for the tutorials, and the package of two disks to save the many hours of typing in lengthy programs.

Two-Disk Package, \$31

.....  
☐ Please send me The Complete Rainbow Guide To OS-9 for \$19.95.\*  
.....  
☐ Please send me The Rainbow Guide To OS-9 Disk (a package of two disks) for \$31.\* Does not include book.  
.....  
Name \_\_\_\_\_ Signature \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ ZIP \_\_\_\_\_  
☐ My check in the amount of \_\_\_\_\_ is enclosed.  
☐ VISA ☐ MasterCard ☐ American Express  
Account Number \_\_\_\_\_  
Interbank No. (MC Only) \_\_\_\_\_  
Card Expiration Date \_\_\_\_\_

**Falsoft, Inc., The Falsoft Building, 9529 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059**

\*Add \$1 shipping and handling per book. Canada and Mexico add \$2. All other foreign countries add \$4. Allow 4 weeks for delivery.  
KY residents add 5% sales tax. In order to hold down costs, we do not bill.  
OS-9 is a trademark of Microware Systems Corporation.

```

12110 '
13000 'GET A SCREEN
13010 'ENTER WITH SCREEN-TO-GET IN VARIABLE SWAP
13020 IF SWAP<1 OR SWAP>6 THEN PRINT"ERROR-CAN'T G
ET SCREEN # "SWAP:RETURN
13030 POKE &H7C41, SWAP
13040 X=USR1(X)
13050 RETURN
13060 '
13070 '
14000 'SET/RESET
14010 'ENTER WITH X1,Y1,CO
14020 IF X1<0 OR X1>767 OR Y1<0 OR Y1>383 THEN PRI
NT"ERROR IN SET-PIXEL OUTSIDE SCREEN COORDINAT
ES":PRINT"X1="X1"Y1="Y1:RETURN
14030 IF CO<0 OR CO>1 THEN PRINT"ERROR IN SET-COLO
R MUST BE 0 OR 1.":PRINT"CO="CO:RETURN
14040 POKE &H7C52,CO
14050 POKE &H7C39,INT(X1/256)
14060 POKE &H7C3A,X1-(256*PEEK(&H7C39))
14070 POKE &H7C3B,INT(Y1/256)
14080 POKE &H7C3C,Y1-(256*PEEK(&H7C3B))
14090 X=USR2(X)
14100 RETURN
14110 '
14120 '
15000 'LINE
15010 'ENTER WITH X1,Y1,X2,Y2,CO
15020 IF X1<0 OR X1>767 OR Y1<0 OR Y1>383 OR X2<0
OR X2>767 OR Y2<0 OR Y2>383 THEN PRINT"ERROR IN LIN
E-PIXEL OUTSIDE SCREEN COORDINATES":PRINT"X1="X
1"Y1="Y1"X2="X2"Y2="Y2:RETURN
15030 IF CO<0 OR CO>1 THEN PRINT"ERROR IN LINE-COL
OR MUST BE 0 OR 1.":PRINT"CO="CO:RETURN
15040 POKE &H7C52,CO
15050 POKE &H7C39,INT(X1/256)
15060 POKE &H7C3A,X1-(256*PEEK(&H7C39))
15070 POKE &H7C3B,INT(Y1/256)
15080 POKE &H7C3C,Y1-(256*PEEK(&H7C3B))
15090 POKE &H7C3D,INT(X2/256)
15100 POKE &H7C3E,X2-(256*PEEK(&H7C3D))
15110 POKE &H7C3F,INT(Y2/256)
15120 POKE &H7C40,Y2-(256*PEEK(&H7C3F))

```

```

15130 X=USR3(X)
15140 RETURN
15150 '
15160 '
16000 'INVERT
16010 X=USR5(X)
16020 RETURN
16030 '
16040 '
17000 'PRINTOUT
17010 CLS:INPUT"PRINTER BAUD SET TO 1200";A$:POKE
150,41:X=USR4(X)
17020 RETURN
17030 '
17040 '
18000 'POINT
18010 'ENTER WITH X1,Y1 RETURNS WITH PNT
18020 IF X1<0 OR X1>767 OR Y1<0 OR Y1>383 THEN PRI
NT"ERROR IN POINT-PIXEL OUTSIDE SCREEN COORDINAT
ES":PRINT"X1="X1"Y1="Y1:RETURN
18030 POKE &H7C39,INT(X1/256)
18040 POKE &H7C3A,X1-(256*PEEK(&H7C39))
18050 POKE &H7C3B,INT(Y1/256)
18060 POKE &H7C3C,Y1-(256*PEEK(&H7C3B))
18070 X=USR6(X)
18080 PNT=PEEK(&H7C47)
18090 RETURN
18095 '
19000 CLEAR 200,31800:PCLEAR4:GOTO 10000

```

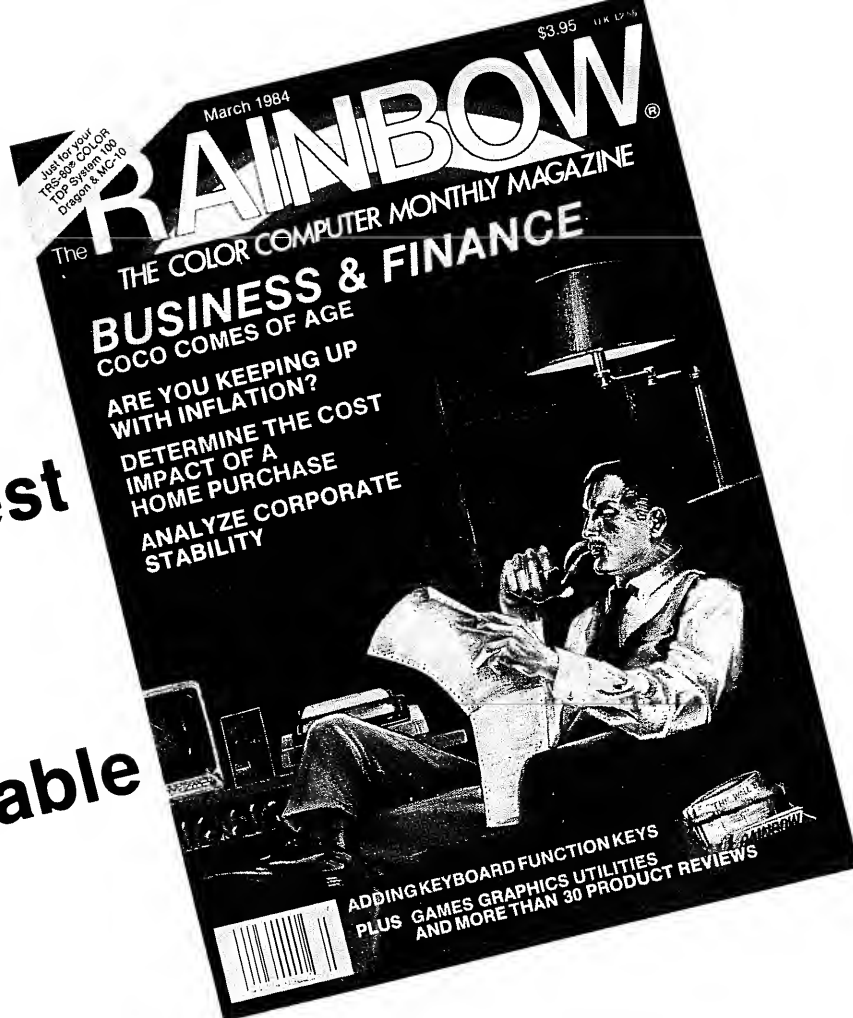
## Listing 2. Disk Patch

```

10211 'DISK PATCH TO MACHINE LANGUAGE
10212 POKE &H7CC7,&H0E: POKE &H7DD7, &H0E: POKE &H
7E55, &H0E: POKE &H7F1F, &H0E: POKE &H7CD2, &H26: P
OKE &H7F28, &H26
11000 'SAVE 6 SCREENS TO DISK
11010 INPUT "FILE NAME";F$:INPUT "DISK READY";A$
11050 SAVEM F$+" "+STR$(SWAP),3584,9727,&H7C71
12000 'LOAD 6 SCREENS FROM DISK
12010 INPUT "FILE NAME";F$:INPUT "DISK READY";A$
12050 LOADM F$+" "+STR$(SWAP)

```

The Biggest  
The Best  
The Indispensable



The latest news on the Color Computer grapevine is that more and more people are discovering **The Rainbow.®**

Now in its third year, **The Rainbow** has become the standard by which other Color Computer magazines are compared. And no wonder! **The Rainbow** towers above the crowd, now offering up to 300 pages each month, including as many as two dozen type-in-and-run program listings, a host of articles and in excess of 30 hardware and software product reviews.

We lead the pack in Color Computer publications and are devoted *exclusively* to the TRS-80® Color, TDP-100 and Dragon-32. We made our climb to the top by continually offering the best and the most by

such well-known authors and innovators as Bob Albrecht and Don Inman, and games from top programmers like Robert Tyson, Fred Scerbo and John Fraysse. **The Rainbow** offers the most in entertainment and education, home uses, technical details and hardware projects, tutorials, utilities, graphics and special features like Rainbow Scoreboard and our CoCo Clubs section.

For only \$28\* a year, you get the keys to all the secrets locked in your CoCo!

Are you searching through the jungle of claims and clamor? Climb above it all. Look up. Find **The Rainbow**.

**THE RAINBOW**  
(502) 228-4492

9529 U.S. Highway 42  
The Falsoft Bldg.  
P.O. Box 385  
Prospect, KY 40059

**THE RAINBOW** is a publication of Falsoft, Inc.

YES! Sign me up for a year (12 issues) of **THE RAINBOW**.

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ ZIP \_\_\_\_\_

Payment Enclosed ☐  
Charge Visa ☐ MasterCard ☐ American Express ☐

My Account# \_\_\_\_\_  
Signature \_\_\_\_\_ Card Expiration Date \_\_\_\_\_

In order to hold down non-editorial costs, we do not bill for subscriptions.



\*Subscriptions to **the RAINBOW** are \$28 (\$31 after Jan. 1, 1985) a year in the United States. Canadian rate \$35 U.S. (\$38 after Jan. 1, 1985). Surface rate to other countries \$65 U.S. (\$68 after Jan. 1, 1985); air rate \$100 U.S. (\$103 after Jan. 1, 1985). All subscriptions begin with the current issue. Please allow up to 5-6 weeks for first copy. U.S. FUNDS only.

Prices subject to change.

# Fie On Intel Hex!

By Stephen P. Allen



Running a "standard" smart terminal (one that lets you download ASCII files using auto-buffer or DC2-DC4 Protocol) on CompuServe (CIS)? Then you're unable to download an assembly language program. You can try, but all you get is an alien stream of meaningless numbers. That bizarre stream of numbers is the program you're trying to download, represented in Intel hex format.

Intelbin will create an A/L tape from a data tape of an Intel hex file. Extended Basic is required to load the resultant A/L tape.

Since the auto-buffer protocol of the standard smart terminal doesn't allow error-checking during actual download, extensive error-checking is provided by Intelbin. (It's assumed

that any error in an A/L file transmission is a fatal one.) If any errors are detected you'll be unceremoniously dumped back to Basic with nothing to show for it.

If your Intel hex data tape is error-free, though, you'll end up with an A/L tape in "gap" format. Those of you with EDTASM+ will recognize this type of tape—there's a silent gap between every data block.

## Non-techies: Skip This!

Basically, there are two formats to consider: the gap-type A/L format, and Intel hex. The gap-type A/L tape stores the program the same way as in a disk file. Instead of having the

load and execution addresses in the name block they're part of the data stream. The actual stream looks like this:

"More" flag	block size	address	data bytes
0:	more	bbbb (two bytes)	load address op codes
\$FF:	no more	0000 (empty)	exec address

As an actual example, consider the short keyboard-scan routine:

```

LOOP JSR $A1C1
    BEQ LOOP
    RTS

```

The opcodes for this routine would be: BD A1 C1 27 FB 39. If this routine were to load and execute at \$5000, the entire gap-type file would be:

```
00 0006 5000 BDA1C127FB39 FF 0000 5000
```

(Spaces inserted for clarity.) The routine would easily fit into one 255-byte tape data block.

Several neat benefits result from storing programs this way, the most important being that the load and execute addresses are part of the data stream. When the program is transmitted, the addresses go along with it. This makes it ideal for telecommunications.

As it happens, this is exactly the same format used by Disk Basic to store its A/L programs on disk, and a lot of disk/tape compatibility headaches are eliminated.

This is fine so far, but if you can't receive 8-bit data from CIS, you can't download that A/L program! Intel hex is the 7-bit solution that CIS has chosen to get around this problem. Basically, each byte is represented by two ASCII hex digits, which can be sent in seven bits each. While this procedure more than doubles transmission time, it does make it possible to receive 8-bit data with a 7-bit terminal program.

Intel hex data is sent in lines that start with a colon and end with a carriage return. They look like this:

```
:nn aaaa 00 (nn data bytes) checksum
```

The last two lines are:

```
:000000
$
```

nn is the number of data bytes in hex (maximum \$18 per line), and aaaa is the load address in hex (not used by Intelbin). The sum of all bytes in the line, including the checksum, must be zero.

If the previous keyboard-scan file were sent in Intel hex, it would look like this (without the spaces):

```
:10 0000 00 0000065000BDA1C127FB39FF00005000 CK (CK = checksum)
```

Note that the entire gap-type file is contained within the date. Intel hex's load address is meaningless in CIS's implementation, and can be ignored, except while computing the checksum.

Note also how the file has grown in size. The six actual opcodes in the routine become 16 in the gap-file, and balloon to 43 in the line above.

### To Use It

To download an A/L program from CIS, go on-line and enter the database. Select the program you want, and command a download. In response to the Protocol question, select DC2-DC4 protocol. CIS may try to stop you at this point, say-

ing, "Protocol selected doesn't support the type of file requested. Continue anyway (Y or N)?" Answer y and you'll get the program in Intel hex format. Save this file to tape as data or Basic ASCII (doesn't matter which, so long as it's a gap-type tape).

When you go off-line, CLOADM and EXEC INTELBIN. Prepare your Intel hex tape to play, and press Enter. Intelbin will scan the tape and convert it to binary.

If there are no errors you'll be instructed to ready an output cassette. You can use the same name as the Intel hex file by simply hitting Enter at the name prompt. To use a new name, type it in and hit Enter.

Now you have an A/L tape you can CLOADM and EXEC. It still loads slowly, and finding the address of the beginning and end points of the program can be a headache. I'll shortly present a short utility called Ungapper that will convert a gap-type A/L tape to the "normal" style single-block file, displaying start, end and exec addresses.

Enjoy!

Note: Intelbin is available in a Basic Loader in the CompuServe Color SIG database xA5 as INTELBI.cc. (end)

### Program. Intelbin

```

00100 *****
00110 ***** INTELBIN *****
00120 *****
00130 * A utility to process *
00140 * M/L files saved in *
00150 * Intel Hex format *
00160 * *
00170
00180 ORG $E00
00190
00200 START CLR $6F (left over from debugging)
00210 JSR $A928 Clear the screen
00220 LBSR MSG1
00230 FCC / INTEL HEX FORMAT DECODER/
00240 FCB /
00250 FCC / FOR binary FILES/
00260 FCB $D0D0
00270 FCC /READY INPUT CASSETTE/
00280 FCB $D0
00290 FCC /AND HIT <ENTER> OR <BREAK>/
00300 FCB 0
00310 WAIT $A1B1 Flash cursor, get keypress
00320 CMPA #3 ...was <break>?
00330 LBEQ EXIT if yes, quit
00340 CMPA #30D ...was <enter>?
00350 BNE WAIT if not, try again

```



0E7B BD	B95C	JSR	\$B95C	print a linefeed	00360	0E7B BD	B95C	JSR	\$B95C	print a linefeed	00360
0E7E 8E	01DA	LDX	#\$1DA	Cassette buffer...	00370	0E7E 8E	01DA	LDX	#\$1DA	Cassette buffer...	00370
0E81 9F	7E	STX	\$7E	is nameblock load address	00380	0E81 9F	7E	STX	\$7E	is nameblock load address	00380
0E83 BD	A701	JSR	\$A701	Load first block: Motorola	00390	0E83 BD	A701	JSR	\$A701	Load first block: Motorola	00390
0E86 1026	008F	IOERR		quit if checksum bad	00400	0E86 1026	008F	IOERR		quit if checksum bad	00400
0E8A 9E	7E	LDX	\$7E	Cass. buffer address to X	00420	0E8A 9E	7E	LDX	\$7E	Cass. buffer address to X	00420
0E8C C6	08	LDB	#8	eight chars. in filename	00430	0E8C C6	08	LDB	#8	eight chars. in filename	00430
0E8E CE	05E2	LDU	\$5E2	Screen address	00440	0E8E CE	05E2	LDU	\$5E2	Screen address	00440
0E91 DF	88	STU	\$88	to Cursor pointer	00450	0E91 DF	88	STU	\$88	to Cursor pointer	00450
0E93 CE	01D2	LDU	#\$1D2	Name also saved for output	00460	0E93 CE	01D2	LDU	#\$1D2	Name also saved for output	00460
0E96 A6	80	LDA	.X+	char. from name	00470	0E96 A6	80	LDA	.X+	char. from name	00470
0E98 A7	C0	STA	.U+	save for later	00480	0E98 A7	C0	STA	.U+	save for later	00480
0E9A BD	A282	JSR	\$A282	Print name char.	00490	0E9A BD	A282	JSR	\$A282	Print name char.	00490
0E9D 5A		DECB		do eight characters	00500	0E9D 5A		DECB		do eight characters	00500
0E9E 26	F6	BNE	NAMLUP		00510	0E9E 26	F6	BNE	NAMLUP		00510
0EA0 0D	7C	TST	\$7C	Is this a namefile block?	00520	0EA0 0D	7C	TST	\$7C	Is this a namefile block?	00520
0EA2 1026	0073	IOERR		quit if not	00530	0EA2 1026	0073	IOERR		quit if not	00530
0EA6 CC	FFFF	LDD	#\$FFFF	Filetype: Ascii, Gaps	00550	0EA6 CC	FFFF	LDD	#\$FFFF	Filetype: Ascii, Gaps	00550
0EA9 A3	01	SUBD	1,X	match?	00560	0EA9 A3	01	SUBD	1,X	match?	00560
0EAB 27	03	BEQ	LAB2	yes	00570	0EAB 27	03	BEQ	LAB2	yes	00570
0EAD 7E	A616	JMP	\$A616	quit if Filemode Error	00580	0EAD 7E	A616	JMP	\$A616	quit if Filemode Error	00580
0EB0 CC	FF01	LDD	#\$FF01		00600	0EB0 CC	FF01	LDD	#\$FF01		00600
0EB3 97	6F	STA	\$6F	Flag cassette input	00610	0EB3 97	6F	STA	\$6F	Flag cassette input	00610
0EB5 D7	78	STB	\$78	Flag "Input" mode	00620	0EB5 D7	78	STB	\$78	Flag "Input" mode	00620
0EB7 0F	79	CLR	\$79	Show file not flushed	00630	0EB7 0F	79	CLR	\$79	Show file not flushed	00630
0EB9 BD	A635	JSR	\$A635	Read 1st tape data block	00650	0EB9 BD	A635	JSR	\$A635	Read 1st tape data block	00650
0EBC 8D	42	BSR	NXTLIN	Read first Intel line	00660	0EBC 8D	42	BSR	NXTLIN	Read first Intel line	00660
0EBE 8D	06	BSR	BINLOD	Go fill the buffer	00670	0EBE 8D	06	BSR	BINLOD	Go fill the buffer	00670
0EC0 BD	A42D	JSR	\$A42D	Close input file	00680	0EC0 BD	A42D	JSR	\$A42D	Close input file	00680
0EC3 16	00B3	LBRA	OUTPUT		00690	0EC3 16	00B3	LBRA	OUTPUT		00690
0EC6 33	8D 01BD	LEAU	*** SMART M/L LOADER ***		00700	0EC6 33	8D 01BD	LEAU	*** SMART M/L LOADER ***		00700
0ECA 8D	2B	BSR	GETDAT	get "More" flag	00730	0ECA 8D	2B	BSR	GETDAT	get "More" flag	00730
0ECC 4D		TSTA		if zero, more to come	00740	0ECC 4D		TSTA		if zero, more to come	00740
0ECD 26	16	BNE	BINEND	if not, go finish up	00750	0ECD 26	16	BNE	BINEND	if not, go finish up	00750
0ECF A7	C0	STA	.U+	store "More" flag	00760	0ECF A7	C0	STA	.U+	store "More" flag	00760
0ED1 8D	1D	BSR	BIN2	get length of file	00770	0ED1 8D	1D	BSR	BIN2	get length of file	00770
0ED3 ED	C1	STD	.U++		00780	0ED3 ED	C1	STD	.U++		00780
0ED5 1F	02	TFR	D,Y	Y gets length of file	00790	0ED5 1F	02	TFR	D,Y	Y gets length of file	00790
0ED7 8D	17	BSR	BIN2	get load address	00800	0ED7 8D	17	BSR	BIN2	get load address	00800
0ED9 ED	C1	STD	.U++		00810	0ED9 ED	C1	STD	.U++		00810
0EDB 8D	1A	BSR	GETDAT	get next opcode	00820	0EDB 8D	1A	BSR	GETDAT	get next opcode	00820
0EDD A7	C0	STA	.U+	and store in buffer	00830	0EDD A7	C0	STA	.U+	and store in buffer	00830
0EDF 31	3F	LEAY	-1,Y	one less opcode to get	00840	0EDF 31	3F	LEAY	-1,Y	one less opcode to get	00840
0EE1 26	F8	BNE	BINLUP		00850	0EE1 26	F8	BNE	BINLUP		00850
0EE3 20	E5	BRA	BINL1	go get next "More" flag	00860	0EE3 20	E5	BRA	BINL1	go get next "More" flag	00860
0EE5 A7	C0	STA	.U+	"More" flag into buffer	00870	0EE5 A7	C0	STA	.U+	"More" flag into buffer	00870
0EE7 8D	07	BSR	BIN2	get empty block length	00880	0EE7 8D	07	BSR	BIN2	get empty block length	00880
00960	C1	00960	STD	.U++	00960	00960	C1	00960	STD	.U++	00960
00970	03	00970	BSR	BIN2	00970	00970	03	00970	BSR	BIN2	00970
00980	C4	00980	STD	.U	00980	00980	C4	00980	STD	.U	00980
00990		00990	RTS	from BINLOAD	00990	00990		00990	RTS	from BINLOAD	00990
01000		01000	BSR	BIN3	01000	01000		01000	BSR	BIN3	01000
01010	BIN2	01010	BSR	GETDAT	01010	01010	BIN2	01010	BSR	GETDAT	01010
01020	BIN3	01020	BSR	EXG	A,B	01020	BIN3	01020	BSR	EXG	A,B
01030	89	01030	EXG	A,B	01030	01030	89	01030	EXG	A,B	01030
01040		01040	RTS		01040	01040		01040	RTS		01040
01050		01050			01050	01050		01050			01050
01060		01060			01060	01060		01060			01060
01070	** Decode data from Intel line **	01070	** Decode data from Intel line **		01070	01070	** Decode data from Intel line **	01070	** Decode data from Intel line **		01070
01080		01080	GETDAT	GETBYT decode next byte from line	01080	01080		01080	GETDAT	GETBYT decode next byte from line	01080
01090	40	01090	DEC	LINLEN,PCR	01090	01090	40	01090	DEC	LINLEN,PCR	01090
01100	8D 0148	01100	BEQ	NXTLIN if line flushed, read next	01100	01100	8D 0148	01100	BEQ	NXTLIN if line flushed, read next	01100
01110	01	01110	RTS		01110	01110	01	01110	RTS		01110
01120		01120			01120	01120		01120			01120
01130		01130	PSHS	B,A save last data	01130	01130		01130	PSHS	B,A save last data	01130
01140	NXTLIN	01140	JSR	\$A176 get next byte from cassette	01140	01140	NXTLIN	01140	JSR	\$A176 get next byte from cassette	01140
01150	NXTL1	01150	CMPLA	#10 ... is linefeed?	01150	01150	NXTL1	01150	CMPLA	#10 ... is linefeed?	01150
01160	0A	01160	BEQ	NXTL1 if so, skip	01160	01160	0A	01160	BEQ	NXTL1 if so, skip	01160
01170	F9	01170	CMPLA	#1: ... is start of Intel line?	01170	01170	F9	01170	CMPLA	#1: ... is start of Intel line?	01170
01180	3A	01180	BEQ	NXTL2 if yes, proceed	01180	01180	3A	01180	BEQ	NXTL2 if yes, proceed	01180
01190		01190	CMPLA	#1\$ ... is end of Intel file?	01190	01190		01190	CMPLA	#1\$ ... is end of Intel file?	01190
01200	24	01200	BEQ	NXTLX if yes, return: no more data	01200	01200	24	01200	BEQ	NXTLX if yes, return: no more data	01200
01210	26	01210			01210	01210	26	01210			01210
01220		01220			01220	01220		01220			01220
01230	* See if last line was short	01230	* See if last line was short		01230	01230	* See if last line was short	01230	* See if last line was short		01230
01240	* If so, then it was final line	01240	* If so, then it was final line		01240	01240	* If so, then it was final line	01240	* If so, then it was final line		01240
01250		01250			01250	01250		01250			01250
01260	18	01260	LDB	#\$18 length of full Intel line	01260	01260	18	01260	LDB	#\$18 length of full Intel line	01260
01270	8D 012F	01270	CMPLB	LBLEN,PCR test against last line	01270	01270	8D 012F	01270	CMPLB	LBLEN,PCR test against last line	01270
01280	1E	01280	BNE	NXTLX if last line short, OK	01280	01280	1E	01280	BNE	NXTLX if last line short, OK	01280
01290	28	01290	IOERR	#\$28 ?IO ERROR code	01290	01290	28	01290	IOERR	#\$28 ?IO ERROR code	01290
01300	AC46	01300	JMP	\$AC46 abort, return to Basic	01300	01300	AC46	01300	JMP	\$AC46 abort, return to Basic	01300
01310		01310	LEAX	LINBUF,PCR Intel line forms here	01310	01310		01310	LEAX	LINBUF,PCR Intel line forms here	01310
01320	8D 0125	01320	JSR	\$A176 get next byte from cassette	01320	01320	8D 0125	01320	JSR	\$A176 get next byte from cassette	01320
01330		01330	TST	Cassette buffer flushed?	01330	01330		01330	TST	Cassette buffer flushed?	01330
01340	70	01340	BNE	IOERR if yes, abort	01340	01340	70	01340	BNE	IOERR if yes, abort	01340
01350	F0	01350	STA	.X+ store in line buffer	01350	01350	F0	01350	STA	.X+ store in line buffer	01350
01360	80	01360	CMPLA	#\$0D carriage return ends line	01360	01360	80	01360	CMPLA	#\$0D carriage return ends line	01360
01370	OD	01370	BNE	NXTL3 keep going till C/R	01370	01370	OD	01370	BNE	NXTL3 keep going till C/R	01370
01380		01380	BSR	CHECKS do checksum, store linelength	01380	01380		01380	BSR	CHECKS do checksum, store linelength	01380
01390	24	01390	LEAX	LINBUF,PCR restore X at start of line	01390	01390	24	01390	LEAX	LINBUF,PCR restore X at start of line	01390
01400	8D 0112	01400	LEAX	8,X skip Intel internal data	01400	01400	8D 0112	01400	LEAX	8,X skip Intel internal data	01400
01410	08	01410	PULS	A,B,PC	01410	01410	08	01410	PULS	A,B,PC	01410
01420	86	01420			01420	01420	86	01420			01420
01430		01430			01430	01430		01430			01430
01440		01440			01440	01440		01440			01440
01450		01450			01450	01450		01450			01450
01460		01460			01460	01460		01460			01460
01470	* Convert two Ascii bytes	01470	* Convert two Ascii bytes		01470	01470	* Convert two Ascii bytes	01470	* Convert two Ascii bytes		01470
01480	* to one binary byte	01480	* to one binary byte		01480	01480	* to one binary byte	01480	* to one binary byte		01480
01490		01490	GETBYT	GETNYB get first byte, convert	01490	01490		01490	GETBYT	GETNYB get first byte, convert	01490
01500	0B	01500	BSR	ASLA	01500	01500	0B	01500	BSR	ASLA	01500
01510		01510	ASLA	shift into	01510	01510		01510	ASLA	shift into	01510
01520	48	01520	ASLA	upper four bits	01520	01520	48	01520	ASLA	upper four bits	01520
01530	48	01530	ASLA		01530	01530	48	01530	ASLA		01530
01540	48	01540	PSHS	A save...	01540	01540	48	01540	PSHS	A save...	01540
01550	34	01550			01550	01550	34	01550			01550



# Off Color: Gampak 3-2-1

By Larry Cadman

We have all heard of Lotus 1-2-3, and Symphony, and recently Softlaw introduced their Workbench, including VIP-writer and Calc. Now a new approach from "QUIXMIX" software is on the scene.

Picture this: you have just finished balancing your check-book, filing your latest recipes, updating your Christmas card list and writing an assembly language graphics arcade game. Now you feel like relaxing and playing a game. Trouble is . . . what game? With Gampak you really don't have to worry, because this is an interactive group of games which includes, among others, versions of a good Pacman, a very exciting Space Invaders, and a cute Donkey Kong. There is also a reasonable implementation of Defenders.

Suppose you go with Kong and then switch to Invaders? You don't have to give up the accumulated score; all score files, including extra men, are transferable between games.

This package has all the features you would expect from state-of-the-art software, including pull-down menus, windowing (used to play two games simultaneously), and icons; and the software supports a mouse.

One of the more interesting features is that if you lose at Defenders you are automatically transferred to the flight simulator (for practice at flying). Also, if you lose two games in a row, you are transferred to Eliza, who analyzes your reasons for wasting your time playing computer games.

The software requires 4K for the tape version or 64K and OS-9 for the disk version. Although the software is copy protected, the author supplies a game which lets you attempt to crack the protection. (This may prove to be the most popular program.)

All sections are written in lo-res assembly language, and the action is smooth and fast. Although the joystick response is somewhat sluggish, this is more than made up for by the rapid transfer between various programs (except in the tape version). Obviously, all score and other files are easily edited by loading into a standard word processor.

We strongly recommend this package for die-hard game addicts. At \$475 the price is a bit steep, but when compared with other packages of the genre (Lotus, etc.) it appears very reasonable. (end)

## UNDERCOLOR

**Finally, a newsletter that will  
keep you up to date on the  
latest Color Computer developments.**

**SUBSCRIBE TODAY!**

- ☐ Yes, enter my subscription to UNDERCOLOR
- ☐ 1 year (26 issues) \$33 (first class mail, \$38)
- ☐ 2 years (52 issues) \$66 (first class mail, \$68)

Canada/Mexico

- ☐ 1 years \$47.00      ☐ 2 years \$94.00

Overseas

- ☐ 1 year \$53.00 US dollars
- ☐ 1 year Airmail \$65 US dollars

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

**Mail to: Color Plus, Box 6809, Roxbury, VT 05669  
or call (802)485-6440**





# Compress That Format!



By Richard Ramella

In *Going Ahead With Color Basic* there is but passing mention of an alternate method of saving programs to tape. Instead of typing `CSAVE "PROG"` to save the material as Basic code, type `CSAVE "PROG",A` to record it as ASCII code.

One problem: the manual doesn't say why anyone would want to save material in ASCII; after all, ASCII saves and loads much more slowly than Basic code.

Discovering that I could deal with programs in ASCII form problems that had been plaguing me for months instantly evaporated. ASCII is more malleable than Basic code; it can be dealt with as a series of string variables, which a common Basic program can work like clay. No machine code or assembly language is needed.

I'd like to share two utility programs which would be worthless without the ability to save material in ASCII form.

The minor example is called "List Formatter," and is used to print program listings in any width you specify, up to the character width limit of your line printer. The major program

is "Professor Compressor." It is an editing utility with several interesting features. It automatically compresses a program by eliminating unneeded spaces and lopping off material rightwards of the remarks lines. It has a search function which lets you find all occurrences of any named character group. And, it lets you append—combine lines. Put all these features together and you have a fairly good program editor which emphasizes shorter listings. Compressed listings run faster and take fewer run bytes—about 80 percent of an uncompressed program.

First you need a Basic program. I suggest you key in the program "Format" in this article. It's brief, gives quick results, and we'll be using it soon anyway. Once Format is resident in your CoCo, put a fresh tape in the cassette recorder, advance the tape past the leader; depress the record and play buttons and type `CSAVE "FORMAT",A` and press Enter. You will notice the material is saved much more slowly than if you had commanded a Basic save. Basic code is highly compressed, contrasted to ASCII code.

The ASCII version will CLOAD as a Basic version. Following your save by SKIPF "FORMAT", followed by the OK prompt, proves a correct load. Now kill the program from the computer; then rewind the tape to the start of the program, type CLOAD "FORMAT" and press Enter. The program returns to the computer.

That's the routine for saving material as ASCII code and reclaiming it as Basic.

We now enter the Interesting Things Department. At the moment you have two versions of "Format"—in Basic in your computer, and in ASCII on tape. Rewind the tape to the start of the saved ASCII material. We're going to have Basic Format print ASCII Format to your specifications.

Run the program in the computer; it advises you to turn on the printer. Next comes a prompt: NAME OF PROGRAM TO BE LISTED? Type FORMAT and press Enter. Then another prompt: NUMBER OF CHARACTERS IN THE LINE? As an experiment, answer this prompt with 20 and press Enter. It will print the program listing in a maximum 20-character width. The program notes: SEARCHING.

It's now time for an onerous task: keying in the program "Prof."

You must save this Basic program in ASCII; let's make it easy on ourselves and use "Format."

"Prof" clears 8000 bytes in line 150; your 16K CoCo only has 8,487 bytes available to contain the rather long listing. Allay your fears: lines 130 and 140 forego Extended Color Basic graphics, not needed in this text-only program, and the available bytes jump to nearly 13,000. There's plenty of room.

Rewind the "Format" ASCII version to its start. Run "Prof" and answer the two prompts: To NAME OF PROGRAM TO BE COMPRESSED?, enter FORMAT. To NUMBER OF LINES IN PROGRAM?, enter 40. Yes, I know *Listing Format* only has 28 lines; claiming about 10 extra lines does no harm; claiming too few will. The lesson here is that you should know how many lines are in the program with which you'll be working. "Prof" will dimension a string array to the number you claim.

The screen will announce: SEARCHING. When "Format" is found the screen blanks and the announcement NOW LOADING FORMAT appears.

Check the lines on screen and you will see how some new things have occurred: only the first entire remarks line is intact. The second remark line has been shortened. Spaces have been eliminated. There are times when spaces should be retained, and "Prof" is equipped to deal with the necessity.

The lines printed on-screen have line numbers and array numbers. After the lines are all printed and assigned array space, you will see a menu with several selections: [S]EEK [L]IST [A]PPEND [P]RINT [C]SAVE. If you do not have a printer, don't attempt to Print.

- **Seek**—Tap keyboard letter s; you will be prompted: SEARCH FOR CHARACTERS? Type the sought character(s) and press Enter. An up arrow will point at the first instance of the characters you entered in the first line of your program. Enter produces the next line containing the characters. SEARCH COMPLETE appears when finished.

- **List**—Press L to print on the screen a list of program lines starting and ending with the lines you choose. Note that the material is identified by its array number, not its line

number. Example: Line 130 of the program may be stored in the A\$(4) array, so it will be displayed in this way: 4 / 130 CLEAR 500. After each request you will be given a prompt: CHOOSE [E]XIT [C]ONTINUE. The first returns to the menu, the second continues with another list.

- **Append**—Combines two sequential program lines. You are asked to enter the array number of the line to which the next line will be added. It won't work if the new line exceeds 255 characters. Also, be sure you want to append the lines because this move cannot be undone.

- **Print**—This function line prints program lines according to your starting and ending specifications. Don't try it if you don't have a printer.

- **Csave**—Press c only when ready to put the newly edited program on tape. You will find an escape clause here if you change your mind. The save will be another ASCII version of the shortened program. It can be loaded as a Basic program and then saved again on tape in a faster-loading Basic version.

To use "Prof" to shorten "Format":

1. Save the Basic program on tape as ASCII code.

2. Run "Prof."

3. Load the ASCII-based program as arrays into "Prof." The result is a series of array strings with nonessential spaces removed and remark lines shortened after the first occurrence.

4. And now for the long instruction: Starting from the beginning of the program, you will determine which lines may be legally appended:

- No remark line may have another line added to it.

- Never append another line to a line with an If-Then test in it.

- Lines which are objective of GOSUB, GOTO, ELSE and THEN must be retained. In other words, if there is a GOSUB 1000 in the program, appending line 1000 to line 990 will create a program crash.

Using the Seek function is a quick way to determine if one line may be appended to the other. Look at the listing for "Format" as an example. Lines 100 and 110 remain as they are because they are remark lines.

Try the Seek function to find the number 120. It shows up only in line 120. Now Seek line 130. It shows up only on line 130. Because no other line branches to line 130, it may be appended to line 120. Now try the List function and look at arrays 1–5. Note that array 4 is line 120 and 5 is line 130. Now use the Append function. Answer the prompt with 4 and you will see the result. Line 120 has had the line 130 material appended to it with the required color between them.

If you try the List function again you will note the arrays have tightened up to allow for the absence of the former 5 array. If you continue the process you may continue appending lines to array 4 so that old lines 120 to 240 are all tightly placed in a single line 120. In fact, 207 characters have been put in the line.

Line 250 must be left as it is because it contains an if-Then test and is also the objective of line 350; and so on.

The result is a tightly-packed program.

If you want to redo anything in the program, don't worry. Use the Save function to put the version on tape, then CLOAD that as a Basic program and edit as usual.(end)

Program. Formatter

```

100 REM * LISTING FORMATER * TRS-80 EXTENDED COLOR
    BASIC / RICHARD RAMELLA
110 REM * REQUIRES LINE PRINTER *
120 CLS
130 CLEAR 500
140 PRINT "TURN ON THE LINE PRINTER."
150 PRINT
160 INPUT "NAME OF THE PROGRAM TO BE LISTED";F$
170 PRINT
180 INPUT "NUMBER OF CHARACTERS IN THE LINE";W
190 PRINT
200 PRINT "SEARCHING..."
210 FOR T=1 TO 500
220 NEXT
230 OPEN "I",#-1,F$
240 CLS
250 IF EOF(-1) THEN 360
260 INPUT #-1,T$
270 FOR X=1 TO LEN(T$)
280 P$=MID$(T$,X,1)
290 PRINT #-2,P$;
300 N=N+1
310 IF N=W THEN PRINT #-2,"": N=0
320 NEXT X
330 N=0
340 PRINT #-2,""
350 GOTO 250
360 PRINT "FORMATER HAS DONE IT'S WORK. SEE LINE LI
    STING..."
370 END

```

Program. Professor Compressor

```

100 REM * PROFESSOR COMPRESSOR * TRS-80 EXTENDED C
    OLOR BASIC 16K
110 REM * A BASIC LISTING COMPRESSER / RICHARD RAM
    ELLA *
120 CLS
130 PMODE 0
140 PCLEAR 1
150 CLEAR 8000
160 BL$=STRING$(32,32)
170 IN$="CHOOSE <E>XIT <C>ONTINUE"
180 INPUT "NAME OF PROGRAM TO BE COMPRESSED";F$
190 INPUT "NUMBER OF LINES IN PROGRAM";Z
200 DIM A$(Z)
210 PRINT "SEARCHING..."
220 FOR T=1 TO 500
230 NEXT T
240 OPEN "I",#-1,F$
250 CLS
260 PRINT "NOW LOADING ";F$
270 FOR Q=1 TO Z
280 IF EOF(-1) THEN 420
290 INPUT #-1,T$
300 V=INSTR(T$,"REM")
310 IF Q>2 AND V>0 THEN A$(Q)=LEFT$(T$,V+2): GOTO
    400
320 FOR A=1 TO LEN(T$)
330 B$=MID$(T$,A,1)
340 IF B$=CHR$(34) THEN S=S+1
350 IF S=2 THEN S=0
360 IF S=1 THEN 380
370 IF B$=CHR$(32) THEN 390
380 A$(Q)=A$(Q)+B$
390 NEXT A
400 PRINT A$(Q)
410 NEXT Q
420 CLOSE #-1

```



```

430 CLS
440 PRINT "MENU"
450 PRINT " <S>EEK <L>IST <A>PPEND <P>RINT <C>SAVE

460 CH$=INKEY$
470 IF CH$="S" THEN GOSUB 640: GOTO 430
480 IF CH$="L" THEN GOSUB 880: GOTO 430
490 IF CH$="A" THEN GOSUB 1020: GOTO 430
500 IF CH$="P" THEN GOSUB 1270: GOTO 430
510 IF CH$="C" THEN PRINT: INPUT " ARE YOU SURE YO
U WANT TO SAVE <Y>ES <N>O";HC$: IF HC$="Y" THEN 530
ELSE 430
520 GOTO 460
530 PRINT "TO SAVE COMPRESSED PROGRAM,"
540 PRINT "DEPRESS PLAY-RECORD BUTTONS"
550 PRINT "ON CASSETTE AND PRESS ENTER"
560 INPUT K
570 OPEN "O",#-1,F$
580 FOR Q=1 TO Z
590 PRINT #-1,A$(Q)
600 NEXT Q
610 PRINT "SEE TAPE. CLOAD"CHR$(34);F$;CHR$(34)
620 PRINT "THIS ENDS THE WORK OF PROFESSOR COMPRES
SOR."
630 END
640 CLS
650 PRINT "SEARCH FOR CHARACTERS"
660 INPUT S$
670 PRINT
680 PRINT
690 FOR Q=1 TO Z
700 SK=INSTR(A$(Q),S$)
710 IF SK=0 THEN 840
720 PRINT @ 192,Q"/ "A$(Q)
730 W$=STRING$(LEN(A$(Q)),"-")
740 FOR XX=1 TO LEN(A$(Q))-LEN(S$)
750 IF MID$(A$(Q),XX,LEN(S$))=S$ THEN MID$(W$,XX,1
)=CHR$(94)
760 NEXT XX
770 PRINT Q"/ "W$
780 PRINT
790 PRINT IN$

```

```

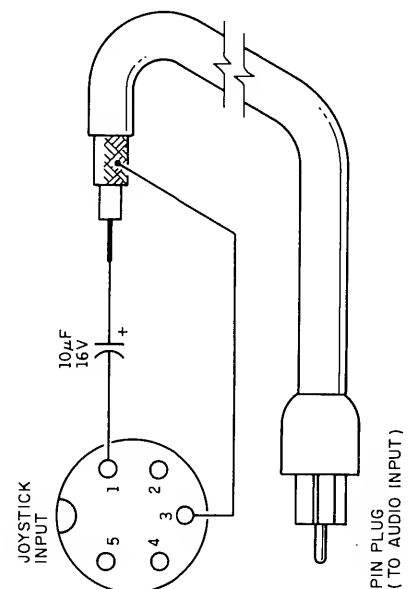
800 CH$=INKEY$
810 IF CH$="E" THEN CLS: RETURN
820 IF CH$="C" THEN CLS: PRINT "SEEK: "S$: PRINT:
GOTO 840
830 GOTO 800
840 NEXT Q
850 PRINT "SEARCH COMPLETE. PRESS A KEY TO RETURN
TO MENU MODE"
860 CH$=INKEY$
870 IF CH$="" THEN 860 ELSE RETURN
880 CLS
890 PRINT "TO SCREEN PRINT LINES."
900 GOSUB 1430
910 IF IM=1 THEN 880
920 FOR LI=ST TO EN
930 PRINT LI"/ "A$(LI)
940 NEXT LI
950 PRINT
960 PRINT IN$
970 CH$=INKEY$
980 IF CH$="E" THEN RETURN
990 IF CH$="C" THEN 880
1000 GOTO 970
1010 IF IM=1 THEN 880
1020 CLS
1030 PRINT "APPEND TWO SEQUENTIAL LINES."
1040 INPUT "ENTER FIRST LINE'S ARRAY NUMBER";NR
1050 NR=INT(NR)
1060 IF NR<1 OR NR>Z-1 THEN PRINT "IMPOSSIBLE": GO
TO 1040
1070 FOR J=1 TO 6
1080 PO=ASC(MID$(A$(NR+1),J,1))
1090 IF PO>57 OR PO<48 THEN RI$=RIGHT$(A$(NR+1),LE
N(A$(NR+1))-J+1): GOTO 1110
1100 NEXT J
1110 OL=LEN(A$(NR))
1120 AL=LEN(RI$)
1130 IF OL+AL>253 THEN PRINT "NEW LINE TOO LONG":
GOTO 1210
1140 A$(NR)=A$(NR)+"."+RI$: A$(NR+1)="
1150 PRINT A$(NR)
1160 FOR EL=NR+1 TO Z-1

```

```

1170 A$(EL)=A$(EL+1)
1180 NEXT EL
1190 A$(Z)=""
1200 Z=Z-1
1210 PRINT
1220 PRINT IN$
1230 CH$=INKEY$
1240 IF CH$="E" THEN RETURN
1250 IF CH$="C" THEN 1020
1260 GOTO 1230
1270 CLS
1280 PRINT "YOU WILL LINE PRINT LISTING"
1290 GOSUB 1430
1300 IF IM=1 THEN 1360
1310 FOR LI=ST TO EN
1320 PRINT #-2,A$(LI)
1330 NEXT LI
1340 PRINT
1350 PRINT "SEE LINE PRINTER LISTING"
1360 PRINT
1370 PRINT IN$
1380 CH$=INKEY$
1390 IF CH$="E" THEN RETURN
1400 IF CH$="C" THEN 1270
1410 GOTO 1380
1420 END
1430 INPUT " ENTER START,END ARRAY NUMBERS SEPARA
TED BY A COMMA";ST,EN
1440 IF ST<0 OR ST>Z OR EN<0 OR EN>Z OR ST>EN THEN
IM=1 ELSE IM=0
1450 IF IM=1 THEN PRINT: PRINT "IMPOSSIBLE": SOUND
100,10
1460 RETURN
1470 END

```



## Errors and Omissions

OPPS WE GOOFED!

This is Figure 4 of Issue 1 for  
Real Time Clock by Dennis Kitz

* LOCAL * IN * LOS * ANGELES * LOCAL * IN * LOS * ANGELES * LOCAL * IN * LOS * ANGELES *		
<p><b>* SOFTWARE *</b></p> <p>Green Mountain Micro!! Computerware (All) Custom Software (No Disc.) Frank Hogg Labs (OS-9 &amp; Flex) Key Color Software (Key 264K) Tom Mix (All) Moreton Bay (More) Prickly Pear (Colorkit!) Spectral Associates Softlaw (VIP &amp; Colorquest) Speech Systems (All) Sugar Software &amp; (Super) CoCoPro (V2) Computer Systems Center Valley Micro Systems (BBS)</p>	<p align="center"></p> <p align="center"><b>* E.D.C. INDUSTRIES *</b></p> <p align="center"><b>Software • Hardware • Support</b> <b>Educational • Small Business • Games</b></p> <p align="center">VOICE (213) 254-6809 10 A.M.-10 P.M. BBS (213) 258-0640 24 HRS. 300 and 1200 Baud Extended Hours+Call before coming.</p> <p align="center"><b>6130 YORK BOULEVARD</b> (Near North Figueroa St.) <b>POST OFFICE BOX 42718</b> <b>LOS ANGELES, CA. 90050</b></p>	<p align="center"><b>EXCLUSIVELY E.D.C.</b></p> <p><b>*232 RemotePlus*</b> by David Cherin</p> <ul style="list-style-type: none"> <li>✓ Requires RS232 or Similar &amp; 64K</li> <li>✓ Remote Terminal Program</li> <li>✓ No Lost or Garbled Data</li> <li>✓ Error Trapping ✓ Software Clock</li> <li>✓ Disable Break Key ✓ Inkey\$ Support</li> <li>✓ New Terminal Program</li> <li>✓ Conversation Mode</li> <li>✓ All Ext. &amp; Disk Commands</li> <li>✓ 20 Commands Added</li> <li>✓ Perfect for BBS</li> <li>✓ 4 Versions Included: Ext., 1.0,1.1 &amp; JDOS</li> <li>✓ Much More - Uses 4K</li> </ul> <p><b>\$24.95 + 2.50 S &amp; H</b> (CA. res. + tax)</p>
* LOCAL * IN * LOS * ANGELES * LOCAL * IN * LOS * ANGELES * LOCAL * IN * LOS * ANGELES *		

## Review— Spectrum DOS

By Jeffrey S. Parker

**PRODUCT**  
Spectrum DOS  
Spectrum Products  
P.O. Box 21272  
Woodhaven, NY 11421  
718-441-2807  
\$49.95 disk  
Requires 64K RAM,  
1 disk drive

Is it not incredible in the microcomputer world of 1985 that when you want lowercase for the Color Computer you are presented with something like an advertisement for Checkerboard Farms? Are you frustrated by the ability to use just one kind of disk drive? Do you weary of typing `LOADM (pause . . .) EXEC?`

If you can answer yes to several or even one of these questions, Spectrum DOS could be for you. Spectrum Disk Operating System can provide the Color Computer with a far greater range of capabilities than RS DOS. Upon receiving the execute command, Spectrum DOS patches into ROM and provides the entire standard command set of the original RS DOS. In addition, it enhances the standard DOS with 24 new commands and several new features and options, such as on-line help and hi-resolution screens.

Table 1 outlines the product's general commands and features. An overview of the new commands and features will give the user a good sense of what Spectrum DOS is, and how it can aid the user and the serious programmer alike with its advanced features.

Spectrum DOS is more interactive with a printer than RS DOS. An echo command allows all output sent to the screen to be also sent to the printer. The memo command gives the user a rudimentary full screen editor, or "scratch pad," for generating memos of notes. A screen dump then sends the note to the printer. A shifted down arrow provides a screen dump at any time, and HDIR will provide a printed copy of an enhanced disk directory.

The error and auto commands will come in handy for programmers; error is "ON ERROR GOTO LINE XXX," and auto initiates an automatic line numbering routine, incremented by tens. Another feature to speed up computing sessions is a device descriptor which lets the user define his disk drives: track per inch, and track to track stepping rate. This means the user can have both sides of a double-sided drive available, and can select the speed at which the drive accesses the disk. Of course, the rate command is limited by the speeds the drive can handle.

While all enhanced features cannot be mentioned here, the hi-res screen is certainly one feature most people would not want to miss. The user is given the option of 32, 51, or 64 character screen widths, with true lowercase. The screen length, however, remains 16 lines.

The RUNM, BOOT//BAS, and DOS commands all provide one-command program loading and running. FLEX will boot Flex, and DOS will boot OS-9. Another handy tool is the set-up file, which lets the user define his disk drives and printer baud, create his own prompt, and define his own cursor. This customization of Spectrum DOS gives the user immediate full use of his system, as he is not required to reconfigure the DOS each time he boots it.

The list continues, but one can get an idea of the many features of this well-designed operating system. One drawback to Spectrum DOS is its EPROM feature. Spectrum DOS can be burned into an EPROM, so that it is immediately available when the computer is turned on, but there are some serious drawbacks to this. The Hi-res, Help, Flip and Flex commands are lost using the EPROM option. While this makes the DOS less attractive to some, others might be willing to make the sacrifice.

Spectrum DOS on the whole is an excellent enhancement for the Color Computer, and should be considered seriously by anyone who is bored, disinterested and aggravated by RS DOS. Tired of slow disk drives and checkerboards? Spectrum DOS could be the "boot" you need! (end)

## Review— Spooler

By Jeffrey S. Parker

**PRODUCT**  
CSPOOL

The Micro Works  
P.O. Box 1110  
Del Mar, CA 92014  
619-942-2400

64K  
disk drive \$19.95 disk  
-or-  
Free with 64K upgrade

Cspool is designed to let the user continue programming while the printer is printing. It is an assembly language program which locates itself in the upper 32K bank of RAM, where it creates its own buffer. It is only activated when device 2 is activated (this being the printer).

Cspool uses the upper 32K of RAM as its buffer. The documentation does not say clearly the amount of actual buffer storage, although it does imply that nearly all 32K are utilized for the buffer. I was unable to fill the buffer when working with the program. Clearing the buffer is very easy; you simply rerun the program. Micro Works provides good documentation with Cspool; it explains how the program runs, states very clearly which software the program will

not work with, and gives a detailed but understandable explanation of how the program functions.

The answer to one of the most frequently asked questions about a spooler program seems to be: Will it work with Telewriter-64? In this case, the answer is no.

The program uses interrupts to send data to the printer while the CPU is not otherwise occupied. These are called foreground and background tasks. Unfortunately, a program like Telewriter-64 uses the same addresses in memory, so this program will not work with that particular word processor. Micro Works suggests that the product be tried with other software to see if it will function; they have been unable to determine what the program will or will not work with.

Micro Works Cspool is an efficient program for generating a temporary buffer in memory and sending information to the printer while the user is working on another task. It is not compatible with known word processors for the Color Computer, but it will function flawlessly when used with any Basic program that does not disturb its memory locations. (end)

## Review— Schematic Drafting Processor

By Jeffrey S. Parker

### PRODUCT

*Schematic Drafting Processor*

*Spectrum Projects*

P.O. Box 21272

Woodhaven, NY 11421

708-441-2807

\$49.95 (disk)

System Requirements: 64K RAM,  
1 disk drive

Up until now, if you wanted to do high resolution schematic drawings you had to use an Apollo DN-660 or a VAX 11/780. Spectrum Projects and Tony "Turn of the Screw" Stefano now bring you the Schematic Drafting Processor. This high resolution graphics processor draws electronic components on a Color Computer monitor screen, and can also dump them to a graphics oriented printer.

What is a schematic drafting processor? Think of a word processor, used to create and edit documents. In this case, drawings of resistors,

capacitors, transistors and diodes, as well as many other electronic components, may be drawn and edited to form very high resolution, accurate schematic drawings.

The Schematic Drafting Processor does this with the use of PCLEAR 24 graphics, or the definition of six overlapping graphics screens to form a 480 x 540 resolution. In addition, the SDP provides on-line help screens for all of its modes, which show the user what options are available in any given drawing mode. There are also 10 optional user definable symbols, so if the more than thirty symbols provided with the SDP are not sufficient, 10 more can be programmed.

Once the schematic drawing has been created, it can be dumped to a printer. SDP will only support three kinds of printers: Radio Shack, Epson, and Gemini. All printers require graphics capabilities, and the Epson requires Grafrax or Grafrax+. Depending on the printer and baud settings, drawings, due to the enormous amount of data being transferred, can take anywhere from 9 to 45 minutes to print.

There are several drawbacks to the SDP which should be mentioned. First, after each character is drawn the program transfers control back to the main command mode; this is the equivalent of typing a word on a word processor, and then having it exit the edit mode into the command mode. SDP would be much easier to work with if after a figure is drawn, the program awaits the next figure to be drawn or a command to return.

Another very important point is understressed in the documentation: The computer must be freshly initialized to run SDP properly. This means the computer must be shut down for at least 15 seconds and restarted prior to loading and running SDP. Just as important, and equally understressed in the documentation, is the error handling routine, which will enable a user to retain the work in progress, even should an I/O error occur during a save or load. The program must be redirected to the main command loop, by typing GOTO 120 if a load error occurs. If a save error occurs, the command GOTO 300 must be typed. Anything else will cause the loss of whatever drawing is in memory at the time. In addition, it may be necessary at that point to restart the computer, as the keyboard may be locked out. While both of these points are mentioned, more attention should be brought to them, to make the points more clearly understood.

Thirdly, while the documentation is clearly detailed and easily understood, there are quite a few typing errors in the tutorial, which include both misspelled and missing words which could mislead or confuse the reader. This is a real detraction from an otherwise well-developed tutorial.

In conclusion, the SDP is a highly advanced program, which competes with equipment hundreds of times its cost. For the engineer, student, hobbyist or hardware developer, SDP could contribute to the quality and efficiency of the work, and could save significantly on time and unnecessary aggravation. If the user frequently has a need to create schematic drawings, the SDP would be an invaluable tool for a software library. (end)

# FEELING HELPLESS? YOU NEED TO LEARN A LESSON!



## MICRO LANGUAGE LAB: LEARNING THE 6809

**F**eeling at the mercy of a programmer somewhere? Mystified by a machine? My **Micro Language Lab** will give you the power to end those helpless feelings.

Here's why. The heart of *any* computer program *anywhere* is machine language. Every piece of software uses it — your favorite game, spreadsheet, word processor, data base manager, or recipe file. Every time you hit "Enter", it's working. All the languages and operating systems are created from it. Basic, Fortran, Pascal, Flex, OS-9,

TRSDOS... they're 6809 machine language, the whole lot!

And you can learn the language of the 6809, the programming heart of your Color Computer, with my **Micro Language Lab**.

Not everyone can program. Writing a few lines in Basic now and then doesn't make you a programmer. But if you *can* program, then my **Micro Language Lab** will teach you — the right stuff, the right way.

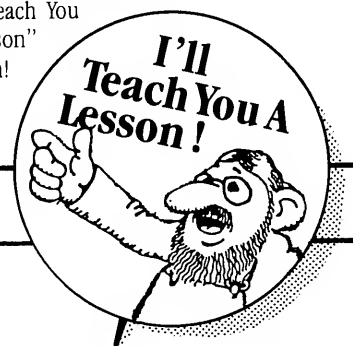
I'll teach you in 24 half-hour lessons on 12 audio cassettes, with a 220-page textbook, with data booklets, with 35 sample programs, and with a programming reference card. You'll spend 50 hours or more with my course, listening, watching, and working. And when

you're done, you'll be programming your Color Computer in the 6809's language.

Stop ignoring those gnawing feelings of helplessness. The 6809 is where the power lies in your computer. Turn on that power with the **Micro Language Lab**.

- **Micro Language Lab**, \$99.00 (plus \$3.50 shipping and handling)  
Requires 16K Extended Basic EDTASM +

Not sure? Write or call for a Table of Contents and sample pages — and your "I'll Teach You A Lesson" button!



## ALSO AVAILABLE FROM GREEN MOUNTAIN MICRO

(Add \$2.50 shipping and handling to your order)

### Lowerkit III

- Full-time upper and lowercase installs in 15 minutes.
- Normal and reverse video standard
- Fully compatible with all Alpha and Graphic modes

\$79.95 assembled and tested

\$49.95 complete kit of parts

*Important! Specify Color Computer or Color Computer II*

**TV Buff II**, clear image for video monitors, \$19.95 (specify CoCo or CoCo2)

**CoCoPort** parallel interface (now includes plastic case), \$54.95 / \$44.95 kit

**ColorPack** ROM/RAM pack, \$29.95 (specify configuration) / \$19.95 kit / case \$3.00 extra

**64K** memory upgrade kit with full instructions, \$49.95 / with memory test on tape, \$54.95

**Color Quaver** Software Music Synthesizer on tape (requires 32/64K), \$19.95

**Color Burner** EPROM Programmer (2716/32/32A/64/64A/128, 68764/66) with software, \$69.95 / \$56.95 kit

User Group, Educational, Club and Dealer Discounts are available.

TRS-80 and TRSDOS are trademarks of Tandy Corporation. Flex is a trademark of TSC, Inc. OS-9 is a trademark of Microware Corporation.

**Green Mountain Micro**  
Bathory Road, Box U  
Roxbury, Vermont 05669  
802 485-6112

Hours: 9am - 5pm EST, Monday - Friday  
COD/VISA/MASTERCARD



# DYNACALC<sup>®</sup>

## CoCo's Best & Fastest Spreadsheet System

### ACCLAIMED BY THE EXPERTS

*"DYNACALC is my choice for a CoCo spreadsheet."*

*Dan Downard, RAINBOW, September, 1984.*

*"Eat your heart out, Lotus 1-2-3!"*

*Scott Norman, HOT CoCo, October, 1984.*

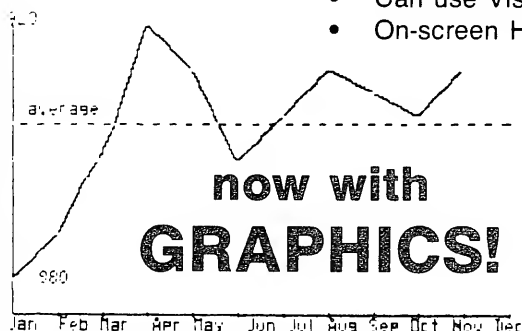
#### Built-in Features:

- 51 x 24 Display with Lower Case
- Super-fast Smart Screen Refresh
- Auto-Repeat Keyboard Driver
- Keysaver (TM) repeats last command x times
- Disk Operating System (works just like ROM DOS)
- Easy communication with BASIC/DOS programs
- Two-way communications with PRO-COLOR-FILE ★Enhanced★
- Outputs to ASCII Word Processors like Telewriter-64
- Fast 16-Digit Arithmetic with Scientific Functions
- Summation, Mean, and Standard Deviation Functions
- Logical Functions with String & Numeric Comparison
- String locate command to navigate large worksheets
- Sort full or partial worksheet by columns or rows
- Line, Bar, Hi/Lo/Close, Circle Graphs
- Full Graphics captioning and overlay facility
- Graphics Drivers for all popular Printers
- Joystick/Mouse Driver for Cursor Movement
- Works with any ROM versions, even JDOS
- 33k Available Worksheet Space
- Up to 256 Columns or 256 Rows
- Can use VisiCalc worksheets & training materials
- On-screen Help Messages



**NOW  
ONLY**

**\$99.95**



#### FOR 64K DISK SYSTEMS

*available from*

#### COMPUTER SYSTEMS CENTER

42 Four Seasons Center #122  
Chesterfield, MO 63017 USA  
(314) 576-5020

10 to 6 Mon.-Fri.

or your local DYNACALC dealer

Price US\$99.95

Outside North America add \$10 postage

DYNACALC Reg. U. S. Pat. Off.

Lotus and 1-2-3 are trademarks of Lotus Development Corp.

PRO-COLOR-FILE is a trademark of Derringer Software

Telewriter-64 is a trademark of Cognitech

VisiCalc is a trademark of VisiCorp.

CANADA-CDN \$129.95  
DATAMAN INTERNATIONAL  
420 FERGUSON AVE. N.  
HAMILTON, ONT. L8L 4Y9  
(416) 529-1319

AUSTRALIA  
PARIS RADIO  
161 BUNNERONG RD.  
KINGSFORD 2032 NSW  
(612) 344-9111

